



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A  
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**VIZUÁLNÍ ZPĚTNOVAZEBNÍ ŘÍZENÍ PRO  
HUMANOIDNÍHO ROBOTA**

VISUAL SERVOING FOR HUMANOID ROBOT

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Pavel Nedvědický**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Jan Najman**

**BRNO 2019**



# Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	<b>Bc. Pavel Nedvěďický</b>
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	<b>Ing. Jan Najman</b>
Akademický rok:	2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## Vizuální zpětnovazební řízení pro humanoidního robota

### Stručná charakteristika problematiky úkolu:

Robotické manipulátory jsou dnes běžnou součástí ve výrobě i výzkumu. Vzhledem k vysokým požadavkům na přesnost a spolehlivost těchto zařízení je jejich pořizovací cena relativně vysoká. Cílem této práce je tedy vývoj robotického manipulátoru v podobě humanoidního robota se dvěma pažemi. Protože bude robot využíván pouze příležitostně pro výuku a demonstraci úloh z oblasti kinematiky, dynamiky a řízení, je zde kladen důraz zejména na nízké pořizovací náklady, přičemž vysoká přesnost a životnost není vyžadována. Robot bude také sloužit pro vývoj a testování pokročilých metod řízení a interakci s okolím. Práce bude řešena jako týmová s jasným vymezením úkolů pro jednotlivé členy.

### Cíle diplomové práce:

1. Navrhněte a sestavte humanoidního robota se dvěma pažemi.
  - Vyberte vhodné pohony a navrhněte konstrukční řešení paží robota.
  - Pokuste se minimalizovat náklady na komponenty (např. využitím repasovaných dílů).
2. Vyberte vhodnou řídicí a výkonovou elektroniku pro pohony.
3. Vyberte vhodný senzor pro vizuální zpětnovazební řízení robota.
4. Provedte základní otestování vybraného řešení.
  - Volba vhodných značek a jejich umístění na konstrukci.
  - Využití více metod detekce značek.
  - Závislost na okolním osvětlení a dalších externích vlivech.
  - Srovnání přesnosti vizuální zpětné vazby s údaji ze senzorů natočení na kloubech.
5. Na základě poznatků z předchozích testů navrhněte vhodná vylepšení systému a naprogramujte ukázkovou úlohu.

**Seznam doporučené literatury:**

CORKE, Peter I. Robotics, vision and control: fundamental algorithms in MATLAB. Berlin: Springer, 2011. Springer tracts in advanced robotics, v. 73. ISBN 9783642201431.

CORKE, P.I. a S.A. HUTCHINSON. A new partitioned approach to image-based visual servo control. IEEE Transactions on Robotics and Automation. 17(4), 507-515. DOI: 10.1109/70.954764. ISSN 1042296X.

LAZAR, C. a A. BURLACU. Visual Servoing of Robot Manipulators Using Model-based Predictive Control. 2009 7th IEEE International Conference on Industrial Informatics. IEEE, 2009, 2009, , 690-695. DOI: 10.1109/INDIN.2009.5195887. ISBN 978-1-4244-3759-7.

CERVERA E. Visual servoing toolbox. Jaume I University, Castello. 2003.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **Abstrakt**

Práce se zabývá konstrukcí levného robotického manipulátoru, který bude sloužit jako exponát a pro výukové účely. Projekt je týmový, autor na něm spolupracoval se spolužákem. Realizována byla robotická paže se čtyřmi stupni volnosti. Pro celého robota byla vybrána řídicí a výkonová elektronika. Autor se zabýval tvorbou softwaru, který je schopen řídit pohony robota pomocí vizuálnímu zpětné vazby, získané zpracováním obrazu z 3D kamery. Na závěr je prezentováno grafické uživatelské rozhraní, které lze využít pro ovládání pohybů robota.

## **Summary**

This thesis deals with construction of a cheap robotic manipulator, which should be used for exhibitions and educational purposes. This project is a teamwork of two students. A robotic arm with four degrees of freedom was developed. Control and power electronics were installed for whole robot. The software's aim is to develop a software that can control the robot by visual feedback, obtained from image processing of an image from 3D camera. Lastly, a graphic user interface for robot movement control is presented.

## **Klíčová slova**

Manipulátor, robot, paletovací robot, Arduino, RealSense, vizuální zpětná vazba, řízení, grafické uživatelské rozhraní, GUI

## **Keywords**

Manipulator, robot, palletizing robot, Arduino, RealSense, visual servoing, regulation, graphic user interface, GUI

NEDVĚDICKÝ, Pavel. *Vizuální zpětnovazební řízení pro humanoidního robota*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/125326>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Jan Najman.



### **Čestné prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím uvedené literatury.

Bc. Pavel Nedvědický, Brno, 2020





### **Poděkování**

Tímto bych chtěl poděkovat všem, kteří mi jakýmkoliv způsobem pomáhali během realizace této práce. Jmenovitě kolegovi Bc. Jakubovi Chlaňovi, se kterým jsem na tomto projektu spolupracoval, strýci Stanislavovi Dvořákovi, který zdarma poskytl použité motory, Ing. Janovi Najmanovi za vedení práce a ochotu rychle a efektivně řešit nastalé problémy. V neposlední řadě děkuji mým blízkým, kteří mě v průběhu realizace této práce i celého studia podporovali.



## Obsah

1. Úvod .....	13
2. Stanovení cílů .....	14
2.1. Rozdělení úkonů .....	14
3. Rešerše tématu .....	15
3.1. Robotické manipulátory .....	15
3.1.1. Paletovací roboti .....	15
3.1.2. Tinkerkít Braccio robot .....	18
3.2. H můstky .....	19
3.2.1. BTS79620B .....	19
3.2.2. Cytron .....	19
3.2.3. L298N Dual H Bridge DC .....	20
3.3. Řídící elektronika .....	20
3.3.1. PLC .....	20
3.3.2. Mikrokontroler s deskou plošných spojů .....	21
3.3.3. Arduino .....	21
3.4. Pohybové senzory .....	22
3.4.1. Senzory Kinect .....	22
3.4.2. Intel RealSense .....	23
3.4.3. Orbbec Astra .....	24
3.5. Zpracování obrazu .....	24
3.5.1. Barevné modely .....	24
3.5.2. Houghova transformace .....	26
3.6. Visual servoing .....	28
3.7. Motion capture .....	29
3.8. Tkinter .....	29
4. Mechanická konstrukce .....	30
4.1. Robotická paže 4 DoF .....	30
4.1.1. Návrh řešení .....	30
4.1.2. Volba pohonů .....	31
4.1.3. Konstrukce .....	32
4.1.4. Efektor .....	35
4.1.5. Shrnutí .....	36
4.2. Robotická paže 6 DoF .....	38
4.3. Základna a tělo .....	39
4.3.1. Základna .....	40
4.3.2. Tělo .....	41
4.3.3. Vertikální posuv .....	41
5. Elektronika .....	43
5.1. Výkonová .....	43
5.1.1. Požadavky .....	43
5.1.2. Cytron MD13S .....	43
5.2. Řídící .....	43
5.2.1. Požadavky .....	43
5.2.2. Arduino Mega 2560 .....	44

5.2.3.	Analogový multiplexer CD74HC4067.....	44
5.2.4.	I <sub>2</sub> C multiplexer TCA9548A.....	45
5.2.5.	Pinout Arduino.....	46
5.3.	Rozvaděč.....	48
6.	Zpracování obrazu.....	49
6.1.	Intel Realsense D435.....	49
6.1.1.	Otestování vybraných funkcionalit.....	49
6.2.	Volba značek.....	50
6.2.1.	QR kód.....	51
6.2.2.	Koule.....	51
6.3.	Umístění značek na konstrukci.....	52
6.3.1.	Levá ruka.....	52
6.3.2.	Pravá ruka.....	53
6.4.	Identifikace značek.....	54
6.4.1.	Čtečka QR.....	54
6.4.2.	Houghova transformace.....	54
6.4.3.	RGB.....	55
6.4.4.	HSV.....	56
7.	Vizuální zpětná vazba.....	58
7.1.	Filtrace dat.....	58
7.2.	Levá ruka.....	59
7.2.1.	Souřadný systém.....	59
7.2.2.	Vzorce.....	60
7.2.3.	Výsledky.....	62
7.3.	Pravá ruka.....	63
7.3.1.	Souřadný systém.....	63
7.3.2.	Vzorce.....	64
7.3.3.	Výsledky.....	67
8.	Řídící algoritmus.....	70
8.1.	Grafické uživatelské rozhraní.....	70
8.2.	Sériová komunikace.....	71
8.2.1.	Formát zprávy.....	71
8.2.2.	Script v Pythonu.....	72
8.2.3.	Script v Arduinu.....	73
9.	Řízení vizuální zpětnou vazbou.....	74
9.1.	Výsledek.....	74
10.	Závěr.....	76
11.	Seznam použitých zkratk.....	77
12.	Zdroje.....	78
13.	Přílohy.....	81
13.1.	Elektronické.....	81
13.2.	Grafické.....	81

## 1. Úvod

Robotické manipulátory jsou v současné době nedílnou součástí průmyslové výroby. Využívají se pro drobné automatizační úkony, jako je například lisování zátek, i ve větších provedeních, například ve výrobních linkách automobilů. Průmyslové verze robotů se dodávají v sestavené podobě a k jejich řízení je využíván software od výrobce. Takový stroj postrádá pro výukové účely jistou míru názornosti. Úkolem této práce je tedy vytvořit manipulátor, inspirovaný průmyslovými řešeními, tak, aby byl využitelný pro demonstrační účely v rámci výuky. Výsledná forma nebyla předem určena, důraz byl kladen pouze na nízké pořizovací náklady. Kromě samotné fyzické konstrukce bude pro robota vybrána a nainstalována vhodná řídící a výkonová elektronika, stejně tak i senzorika.

Po dokončení robota z hlediska mechaniky a elektroniky se tato práce bude zabývat realizací zpětné vazby za pomoci kamery. Údaje určené vizuální zpětnou vazbou se využijí pro řízení robota bez použití jiných údajů o poloze. Na rozdíl od zpětné vazby získané například pomocí enkodérů umožňuje vizuální zpětná vazba získat ucelený přehled o pracovním prostoru, což může posloužit pro detekci překážek, předmětů, přítomnosti osob a další.

Tato práce byla zadána jako týmová, proto autor na mechanické konstrukci a elektronice úzce spolupracoval s kolegou Bc. Jakubem Chlaňem [1]. Jednotlivé konstrukční prvky byly vymyšleny a realizovaly společnými silami. Pro účely psaní závěrečné práce jsou jednotlivé úkony rozděly tak, aby je ve své práci měl zapsány ten, který na nich měl větší podíl.

Software použitý v této práci bude psán v programovacím jazyce Python, jelikož je vhodný pro účely, které vyplývají ze zadání práce. Zároveň se autor chtěl tento jazyk naučit. 3D modely dílů se vytvářely v programu Autodesk Fusion 360.

## 2. Stanovení cílů

Cílem tohoto projektu je návrh a tvorba laboratorního modelu průmyslového robota, a to konkrétně jeho mechanických částí, elektroniky a základního řídicího softwaru. Jelikož jsou jednotlivé body rozsáhlejšího charakteru, byl projekt vypsán jako týmový, přičemž na jednotlivých úkonech budou v různé míře spolupracovat dva studenti.

Robot by měl sloužit jako pomůcka při výuce, kde by našel konkrétní využití při demonstraci kinematických úloh a různých metod řízení. Bylo by možné na něm realizovat semestrální práce do předmětů s odpovídajícím zaměřením, případně robota vylepšovat v rámci bakalářských či diplomových prací. Další možnost využití je při prezentaci oboru mechatroniky na různých dnech otevřených dveří a výstavách.

Během prvotních úvah nad tématem bylo rozhodnuto, že se vytvoří dvě paže, které umožní větší variabilitu pohybů a zároveň budou zajímavější po estetické stránce. Obě paže by se od sebe měly odlišovat konstrukčně, aby každá mohla sloužit k demonstraci jiného typu pohybů a inspirovala se jiným druhem manipulátoru. Jednodušší paže bude mít čtyři stupně volnosti, zatímco ta druhá šest.

### 2.1. Rozdělení úkonů

Většinu úkonů, zejména z mechaniky a elektroniky, řešili oba autoři společně. V jednotlivých pracích jsou popsány ty úkony, které měl na starost daný autor, případně na nich měl dominantní podíl. V této práci je popsána tvorba jedné z rukou, konkrétně té se čtyřmi stupni volnosti. Práce dále zahrnuje konstrukci těla, výběr vhodné výkonové a řídicí elektroniky. Samostatnou činností autora této práce jsou kapitoly o vizuální zpětné vazbě za pomoci kamery, řízení robota pomocí ní a také tvorba grafického uživatelského rozhraní.

Bc. Jakub Chlaň ve své práci popisuje konstrukci druhé ruky se šesti stupni volnosti, pro obě ruce vybírá a implementuje vhodnou sensoriku, potřebnou pro řízení a bezpečnost – získávání zpětné vazby z pohonů, snímače vzdálenosti, proudové senzory, koncové spínače. V softwarové části popisuje dopřednou i inverzní kinematiku a implementuje řídicí algoritmy pro všechny motory v prostředí *Matlab/Simulink*.

### 3. Rešerše tématu

Rešeršní kapitola se zabývá prozkoumáním dílčích problematik, které budou řešeny v následujících kapitolách. Nachází se zde tedy podkapitola o průmyslových robotech, která posloužila jako inspirace pro mechanickou konstrukci. Vybírá se výkonová elektronika, která bude sloužit k ovládání motorů. Je provedena rešerše řídicí elektroniky, umožňující ovládání celého robota, i dostupných senzorů obrazu. V dalších bodech jsou popsány teoretické základy zpracování obrazu, metody vizuálního zpětnovazebního řízení, využití míček pro účely identifikace pohybu objektů a na závěr možnosti tvorby grafických uživatelských rozhraní.

#### 3.1. Robotické manipulátory

Většina průmyslových manipulátorů je vyráběna se šesti stupni volnosti. Vzhledem k rozdělení úkolů (viz 2.1) jsou takové manipulátory náplní práce [1]. Naopak tato práce se zabývá řešením konstrukce robota s nižším počtem stupňů volnosti, tomu odpovídají i rešeršovaná zařízení.

##### 3.1.1. Paletovací roboti

Jedním poddruhem průmyslových robotů jsou roboti označovaní často jako paletovací (angl. palletizing robot). Vyznačují se svojí typickou konstrukcí a uspořádáním pohonů. Najdou v průmyslu uplatnění pro širokou škálu činností, při kterých dochází k manipulaci s materiálem. Najdeme je zejména na konečných stanovištích výrobních linek, kde zakládají hotové výrobky z pásu na paletu, určenou k expedici.



*Obr. 1: Paletovací robot během činnosti*

Paletovací manipulátory se vyrábí ve dvou variantách, odlišujících se od sebe počtem stupňů volnosti – mají buď čtyři, nebo pět. Oba typy jsou charakteristické obdobným rozložením os rotací. První aktuátor, upevněný k základně, rotuje ve vertikální ose. Další tři rotace probíhají v horizontální směru, přičemž osy rotací jsou vůči sobě rovnoběžné. U většiny výrobců lze pořídit i variantu s jedním dodatečným stupněm volnosti, díky kterému lze konečný bod natáčet ve vertikální ose. Rameno je zakončeno efektořem, který slouží k uchopování objektů a jeho podoba závisí na konkrétním uplatnění. Následující body zmiňují konkrétní typy paletovacích robotů od firem FANUC a KUKA, které se používají v průmyslu.

a) FANUC M-410

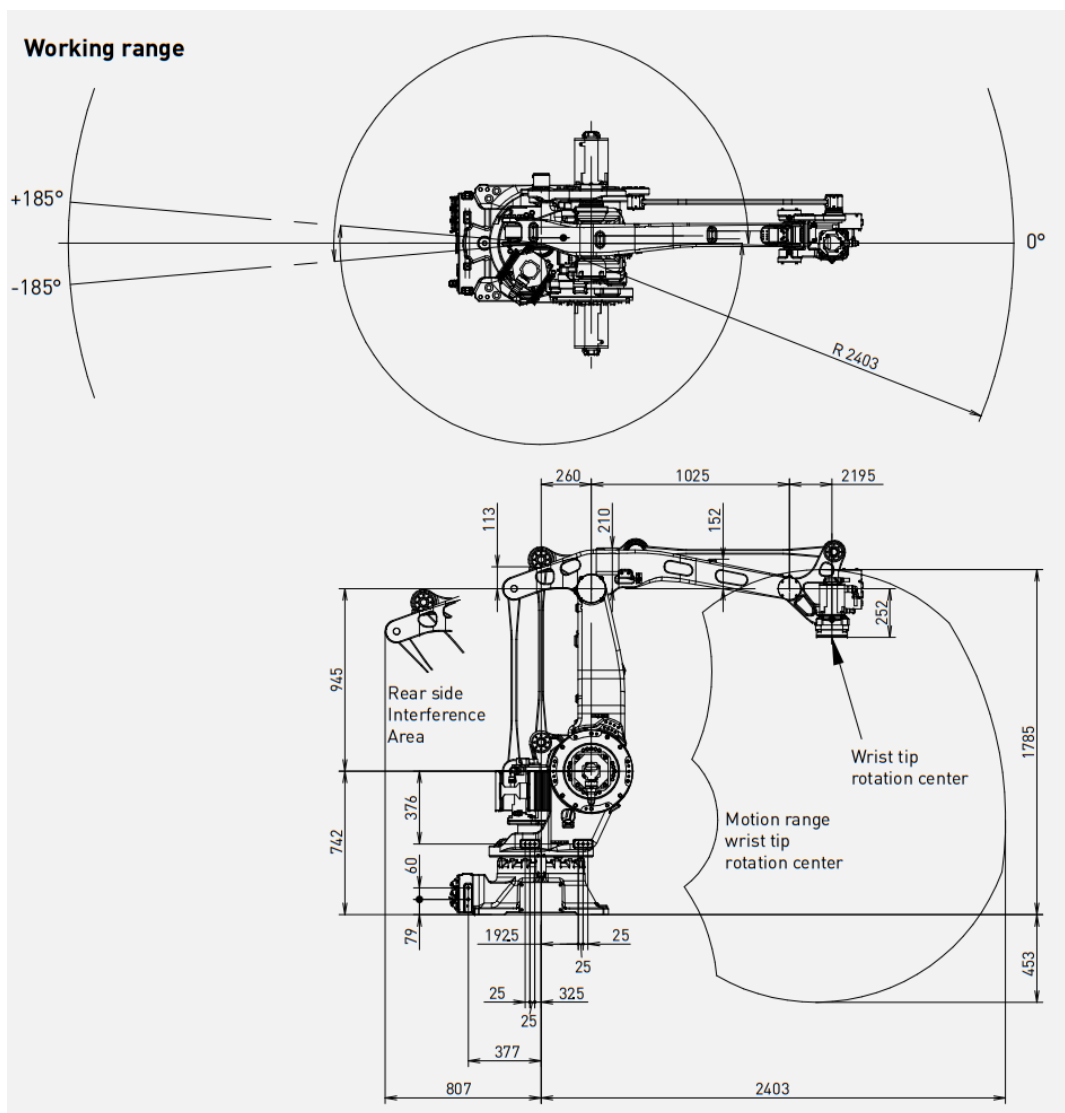
Produktová řada M-410 [2] od Japonské společnosti FANUC, zabývající se řešeními pro průmyslovou automatizaci, se skládá z množství robotů různých velikostí a různých maximálních zatížení. Roboti jsou poháněni servomotory, napájenými sdruženým napětím z elektrické sítě. Jelikož manipulátor, který bude vytvářen, bude podstatně menších rozměrů, je řešerše takových pohonů irelevantní. Cílem je zejména inspirovat se z hlediska uspořádání vazeb a pohonů.

Nejmenší variantou je verze M-410iC/110 (*obr. 2*). Rotace ve vertikální ose je zajišťována pohonem, který je umístěn v základně robota a je vybaven šnekovou převodovkou. Horizontálních rotací je dosaženo pomocí pákového mechanismu, který má za cíl zejména zvýšení nosnosti robota. Pohony tak nemusí být umístěny poblíž os rotace, lze je uložit do základny.



*Obr. 2: FANUC M-410iC/110 [2]*





Obr. 3: FANUC M410iC/110 – schéma [3]

#### b) KUKA QUANTEC

Obdobná řešení má ve svém katalogu i konkurenční Německá firma KUKA [4], která navíc disponuje větší rozmanitostí konstrukcí. Kromě robotů s pákovým mechanismem, podobným výrobkům FANUC, nabízí například řada QUANTEC modely s pohony přímo ve vazbách. Nejkompaktnějším modelem je typ KR 120 R3200 PA (obr. 4). Jeho maximální dosah činí 3195 mm, nosnost je až 120 kg a váží něco přes tunu. Jedná se o variantu s pěti stupni volnosti.



Obr. 4: KUKA KR 120 R3200 PA [5]

### 3.1.2. Tinkerkit Braccio robot

Braccio [6] je manipulátor, inspirující se průmyslovými řešeními, určený pro hobby účely, a to kvůli své přijatelné ceně a kompaktním rozměrům, které umožňují instalaci na stůl. Součástí kitu je i efektor, který je tvořen dvěma packami a soustavou ozubených kol, díky kterým se při pohybu servomotoru efektor zavírá. Vazby jsou poháněny modelářskými servomotory, konkrétně dvěma SpringRC SR431 a čtyřma SpringRC SR311. Většina dílů je z plastu a díky tomu sestava váží jen necelý kilogram.



Obr. 5: Tinkerkit Braccio robot [6]

Parametr	Hodnota
Hmotnost	792 g
Maximální dosah	800 mm
Maximální výška	520 mm
Šířka gripperu	90 mm
Maximální zatížení	150 g

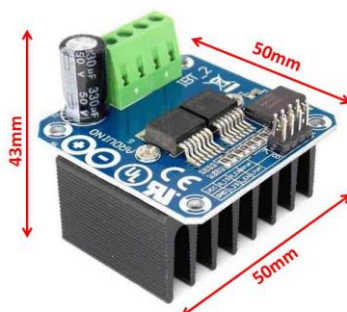
Tab. 1: Parametry robota

### 3.2. H můstky

H můstek je aktivní elektronický obvod, sloužící k ovládání stejnosměrného motoru, a to v obou směrech rotace i obou režimech – motorickém a generátorickém. Jeho základem jsou čtyři tranzistory, jejichž vhodným spínáním se dosáhne požadovaného chování motoru. Rešerše porovnává komerčně prodávané H můstky, dimenzované na 12–24 V, protože v konstrukci budou využity pohony pracující právě na těchto napětích.

#### 3.2.1. BTS79620B

Tento H můstek je navrhnut pro obvody využívající desku Arduino, avšak lze jej použít pro libovolný obvod s odpovídajícími parametry. Je schopen regulovat napětí v rozmezí 6 až 27 V DC a špičkově zvládne průtok proudu o velikosti 43 A. K řízení jsou zapotřebí čtyři piny - 5 V, GND, PWM 1 a PWM 2. Každý PWM pin slouží pro otáčení jedním směrem. Jeho cena se pohybuje okolo čtyř set korun.



Obr. 6: H můstek BTS7960B [7]

#### 3.2.2. Cytron

Firma Cytron nabízí řadu H můstků [8], které se liší podporovaným rozsahem napětí, maximálním kontinuálním a špičkovým proudem, ochranami proti nadproudu, podpětí nebo vysoké teplotě. Některé mají více kanálů, což umožňuje nezávislé řízení většího počtu pohonů. Vhodné varianty a jejich klíčové parametry jsou vypsány v tab. 2.

Model	Napětí [V]	Proud stálý / špičkový [A]	Ochrany	Počet kanálů	Cena [Kč]
FD04A	7–25	1,5 / 2,5	Opačná polarita	4	750
MDD10A	5–30	10 / 30	Ne	2	500
MD13S	6–30	13 / 30	Nadproud, podpětí	1	250
MD20A	6–30	20 / 60	Nadproud, podpětí, přehřátí	1	450

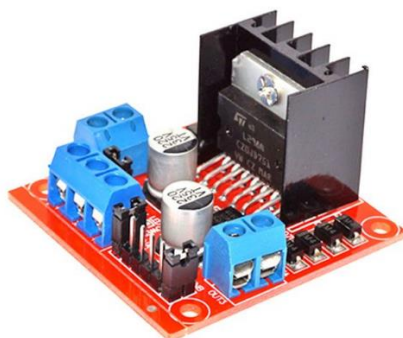
Tab. 2: H můstky Cytron, porovnání



Obr. 7: Cytron MD10C [8]

### 3.2.3. L298N Dual H Bridge DC

Oproti předchozím popsaným výrobkům se tento liší zejména tím, že je schopen řídit krokový motor, případně dva stejnosměrné motory. Operuje s napětími v rozmezí 5 až 35 V a jedna větev může trvale přenášet proud až do velikost dvou ampér, špičkově tři ampéry. Jeho rozměry jsou 55x60x30 mm a jeden kus stojí přibližně 70 Kč.



Obr. 8: L298N [9]

## 3.3. Řídicí elektronika

### 3.3.1. PLC

PLC (angl. programmable logic controller), českým názvem programovatelný logický automat, je průmyslový počítač, který se používá pro řízení výrobních linek a strojů. Je uzpůsoben pro řešení automatizačních úkonů v reálném čase. Interakci s okolím zajišťují moduly, které obsahují vstupní i výstupní periferie. Součástí PLC sestavy je i rozhraní, které slouží pro komunikaci s programátorem a obsluhou [10].

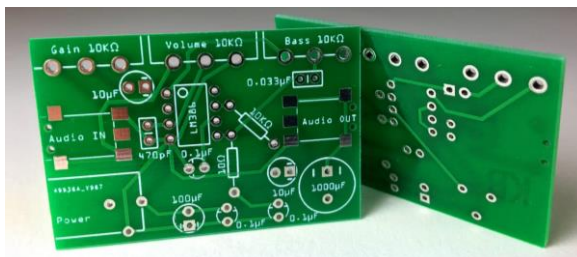
Mezi nejznámější firmy, které se vývojem a výrobou PLC zabývají, patří Siemens, B&R, Schneider. Podrobněji si PLC popíšeme na produktu řady Siemens LOGO!. Srdcem produktu je základní modul, který je řídicím členem pro ostatní. Počet jeho vstupů a výstupů dostačuje pouze elementárním aplikacím, proto se obvykle připojí potřebný počet rozšiřujících modulů. Jako rozhraní pro komunikaci s uživateli se nabízí různé displeje a panely s tlačítky. Pro samotné programování zařízení firma Siemens vyvíjí software LOGO! Soft Comfort, ve kterém je možno grafickou formou tvořit programy [11].



Obr. 9: Základní modul Siemens LOGO! [11]

### 3.3.2. Mikrokontroler s vlastní deskou plošných spojů

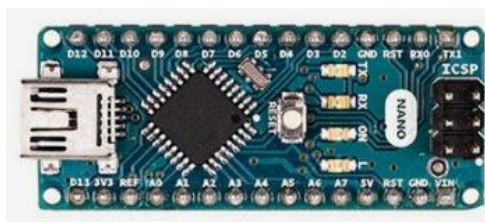
Mikrokontroler je programovatelný jednočipový počítač, tvořený jednoduchým integrovaným obvodem, který se využívá pro různé jednoúčelové aplikace. Vyznačuje se svojí kompaktností a spolehlivostí. Součástmi čipu jsou procesor, paměť, RAM, časovače a další. Mikročip se instaluje na navrhnoutou desku plošných spojů, což je deska určená k osazení elektronickými součástkami, realizujícími elektronické obvody. Osazená deska je po naprogramování čipu schopna plnit uvažované funkce – snímat vstupy, generovat výstupy, komunikovat po sběrnicích....



Obr. 10: Deska plošných spojů [12]

### 3.3.3. Arduino

Arduino [13] je open-source platforma, mezi jejíž klíčové vlastnosti patří jednoduchost, a to po stránce hardware i software. Arduino vzniklo jako nástroj pro fast prototyping, brzy se rozšířilo mezi kutily, mimo jiné kvůli nízké ceně a open source provedení. Velkou výhodou se stala široká komunita, díky níž lze spoustu řešení získat z diskuzních fór a tutoriálů. S Arduinem lze vytvářet projekty různých složitostí, od jednoduchých aplikací, jako například teploměr, až po složitější, mezi které patří právě ovládání robotické ruky. Podle potřebného počtu vstupů a výstupů lze pořídit různé modely Arduina, od modelu Nano po Mega. V nabídce jsou také kity a starter packy, které v krabici obsahují vše, co je potřeba pro snadný začátek.



Obr. 11: Arduino Nano [14]



Obr. 12: Arduino Mega 2560 [15]

### 3.4. Pohybové senzory

Pozdějším cílem této práce je ovládat robota vizuální zpětnou vazbou. K tomu je zapotřebí důkladně vybrat vhodný senzor, který splňuje předurčené požadavky. Za předpokladu správně vytvořeného softwaru jsou to právě vlastnosti senzoru, které budou mít rozhodující vliv na schopnost přesně určit polohu robotické paže. Chceme-li určit polohu objektu v prostoru, je zapotřebí se poohlédnout po senzorech, které kromě obrazu umožňují snímat i hloubku jednotlivých pixelů a tím získat 3D obraz prostoru.

#### 3.4.1. Senzory Kinect

Senzory řady Kinect patří k nejznámějším svého druhu, stojí za nimi Microsoft a od prvního vydání v roce 2010 se na trh dostalo již pět modelů. První, Kinect for Xbox 360, byl určen pro propojení s herní konzolí a díky schopnosti sledovat pohyb osob způsobil revoluci v možnostech hraní her. O Kinect byl velký zájem a s osmi miliony prodanými kusy během prvních šedesáti dní se zapsal do Guinnessovy knihy rekordů. Senzor byl vybaven RGB kamerou, infračerveným hloubkovým senzorem a mikrofonom [16].



*Obr. 13: Kinect for Xbox 360 [16]*

Postupem času zájem o Kinect opadal a Microsoft se rozhodl, že už Kinect jako příslušenství pro Xbox vyvíjet nebude, avšak v březnu 2019 představil Azure Kinect [17], který je prezentován jako sada pro vývojáře. Výrobce jako příklad dává využití ve zdravotnictví, retailu, logistice, případně robotice. K senzoru lze vyvíjet vlastní software nebo využít různé funkce od výrobce, mezi které patří sledování pohybu či rozpoznávání řeči. Záměrem je propojení senzoru s Azure, což je cloudová služba Microsoftu. Senzor je vybaven například následujícími funkcemi:

1. 1megapixelový hloubkový senzor
2. Pole se 7 mikrofony
3. 12megapixelová RGB videokamera
4. Akcelerometr a gyroskop
5. Synchronizační kolíky pro synchronizaci více senzorů



*Obr. 14: Azure Kinect [17]*



### 3.4.2. Intel RealSense

Intel nabízí pod hlavičkou Realsense řadu svých produktů určených k stereovizi. Ty se liší výbavou i cenovkou, proto se nyní podíváme na jejich porovnání. Začneme modelem SR305 [18], který láká svojí cenou okolo dvou tisíc korun. Je vybaven technologií kódovaného světla, díky které je schopen přesně snímat do vzdálenosti jednoho metru. Funguje na principu projekce vzorů do prostoru a na základě deformace těchto vzorů dokáže spočítat hloubku jednotlivých bodů.



Obr. 15: Realsense SR305 [18]

Série D4xx v současné době obsahuje tři modely, D415, D435 a D435i [19]. Hlavním rozdílem mezi D415 a D435 je zorný úhel, který u D415 činí zhruba 65°, zatímco u D435 je to 87°. Důsledkem rozdílného zorného úhlu je větší počet pixelů na stejné ploše v případě D415, což má podstatný vliv na přesnost hloubkového senzoru, která je podle výrobce až dvojnásobná. Verze D435i je oproti D435 navíc vybavena IMU (inertial measurement unit), která umožňuje měřit natočení kamery v prostoru.



Obr. 16: Realsense D435 [20]

Parametr	D415	D435	D435i
Rozměry	99 x 20 x 23 mm	90 x 25 x 25 mm	90 x 25 x 25 mm
<b>Hloubkový senzor</b>			
Technologie	Active IR stereo	Active IR stereo	Active IR stereo
Zorný úhel	65°	87°	87°
Rozlišení	1280 x 720, 90 FPS	1280 x 720, 90 FPS	1280 x 720, 90 FPS
<b>RGB kamera</b>			
Zorný úhel	69,4°	69,4°	69,4°
Rozlišení	1920 x 1080, 30 FPS	1920 x 1080, 30 FPS	1920 x 1080, 30 FPS

Tab. 3: Vybrané parametry senzorů řady D4xx

### 3.4.3. Orbbec Astra

Firma Orbbec nabízí ve své sérii Astra tři modely, a to Astra S, Astra a Astra Pro [21]. Všechny jsou přímými konkurenty výše zmíněných modelů, stejně tak i výrobcem navrhované využití těchto senzorů je obdobné jako v předchozích dvou případech – zdravotnictví, automatizace, robotika, smart buildings... Kamera láká zejména na možnosti svého softwaru, mezi které se řadí rozpoznání obličejů, rozpoznání gest, sledování pohybu člověka a další. Žádný z tří modelů však nemá v ČR svého distributora, což by značně komplikovalo pořízení výrobku této značky.



Obr. 17: Orbbec Astra Pro [21]

Parametr	Astra S	Astra	Astra Pro
Rozměry	165 x 30 x 40 mm		
Zorný úhel	60 x 49,5 x 73°		
RGB kamera	640 x 480, 30 FPS	640 x 480, 30 FPS	1280 x 720, 30 FPS
Hloubková kamera	640 x 480, 30 FPS	640 x 480, 30 FPS	640 x 480, 30 FPS
Dosah	0,4 – 2 m	0,6 – 8 m	0,6 – 8 m

Tab. 4: Vybrané parametry senzorů Orbbec Astra

### 3.5. Zpracování obrazu

Robot bude snímán vybranou kamerou z výše uvedených a bude zapotřebí detekovat jeho části pro účely zjištění jeho aktuální polohy, proto jsou zde popsány některé způsoby, kterými toho lze dosáhnout. Zmíníme zde další důvody, proč bude programováno v Pythonu. Kromě toho, že výše zmíněné kamery mají v Pythonu podporu, tak je další výhodou i to, že Python je jedním z jazyků, které mohou využít knihovnu OpenCV, která patří mezi nejznámější a nejoblíbenější knihovny pro počítačové vidění a zpracování obrazu. Obsahuje většinu běžně potřebných funkcí, mezi které patří transformace mezi barevnými modely, detekce objektů a lidí, hledání podobných obrázků a spousta dalšího. Z jejího názvu je zřejmé, že se jedná o open source knihovnu. Přispívá do ní přes 47 tisíc uživatelů a byla stažena více než 18 milionkrát. Takto rozsáhlá komunita ulehčuje hledání informací o řešeních.

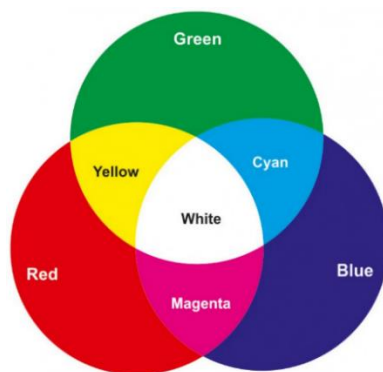
#### 3.5.1. Barevné modely

Pro počítačovou interpretaci barev byly vytvořeny různé barevné modely a ty nejobvyklejší z nich si nyní popíšeme, protože další součástí práce bude i detekce různých objektů a jedním ze způsobů, jak je lze detekovat, je podle jejich barvy. Obecně se barevný model skládá ze složek, přičemž jednotlivé barvy odpovídají kombinaci podílů těchto složek.



#### a) RGB

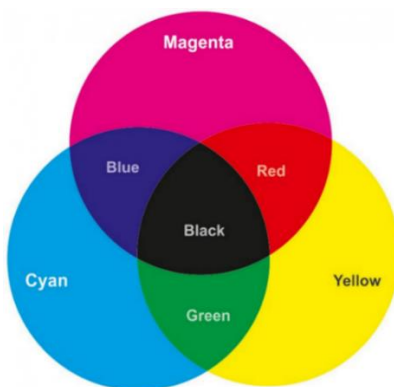
Nejběžněji využívanou reprezentací obrazu je RGB model, který se využívá pro skládání obrazu na displejích. Barva vzniká kombinací třech základních barev – červené (R), zelené (G) a modré (B). Obvykle je každý parametr reprezentován osmibitovou hodnotou z intervalu 0 až 255, odpovídající intenzitě dané složky. Nulová hodnota všech složek odpovídá černé barvě, maximální hodnoty bílé. Tento model je aditivní, to znamená, že výsledná barva se tvoří jako součet příspěvků jednotlivých základních barev.



*Obr. 18: Znáznornění RGB modelu [22]*

#### b) CMY(K)

Narozdíl od RGB, CMYK je modelem subtraktivním. To znamená, že se příspěvky jednotlivých složek odečítají, a tudíž jejich maximální intenzitou vznikne černá. Skládá se ze třech složek – azurové, purpurové a žluté. Často se doplňuje i černá barva, protože tohoto barevného modelu využívají tiskárny a v reálném světě by smícháním tří složek přesná černá nevznikla, jednalo by se spíše o tmavě hnědou.

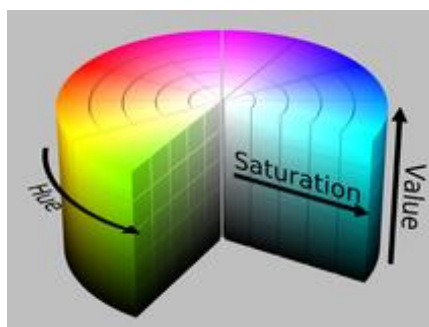


*Obr. 19: CMY(K) barevný model [22]*

#### c) HSV/HSB

Ani jeden z předchozích modelů neodpovídá tomu, jak spektrum barev vnímá lidské oko. Vznikl tedy tento model, který má reprezentaci barev přiblížit vnímání člověka. Je reprezentován třemi složkami, a to barevným tónem ( $H$  = hue), sytostí ( $S$  = saturation). Poslední složka je jas, označena jako  $B$  (brightness), nebo  $V$  (value). Obě označení se liší pouze názvem, skrývá se za nimi totožná vlastnost.

První parametr H reprezentuje tón barvy na 360° kruhu, přičemž každé natočení odpovídá barvě, například 0° je červená, 120° zelená a 240° modrá. Zbylé dva parametry se vyjadřují v procentech. Čím je vyšší sytost S, tím výraznější je daná barva. Parametr V značí jas, nižší hodnoty reprezentují tmavší barvy a vyšší světlejší. Tento barevný profil se často zobrazuje jako válec, kde úhel reprezentuje parametr H, vzdálenost od středu parametr S a výška parametr V. V OpenCV se nevyužívají rozsahy 0°-360° a 0-100 %. H je vyjádřen na škále 0–179, zbylé dva parametry osmibitovou hodnotou 0-255.



Obr. 20: Znázornění HSV modelu [23]

#### d) Grayscale

Černobílý obraz je reprezentací barevného obrazu v odstínech šedi. Pro převod z RGB se využívají různé způsoby. Nejjednodušší z nich je průměr třech složek – (1).

$$Y = \frac{R + G + B}{3} \quad (1)$$

Pokročilejší modely pracují s váženým průměrem jednotlivých barev. Důvodem je odlišná citlivost lidského oka na jednotlivé složky barevného spektra, proto takto převedený obraz působí přirozeněji. Rovnice s váženými průměry, které využívá funkce implementovaná v OpenCV, je sepsána v (2). Existují však i další modely, které mají koeficienty mírně odlišené.

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (2)$$

### 3.5.2. Houghova transformace

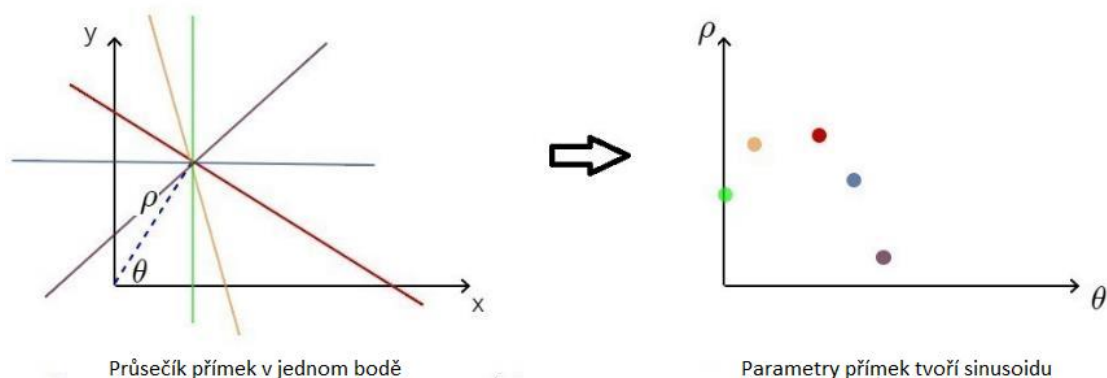
Dalším způsobem, jakým lze objekty v obraze detekovat, je využitím Houghovy transformace. Je to metoda, kterou se dají identifikovat přímky, kružnice i jiné tvary. Ve všech případech se nejprve detekují hrany a poté se hledají vhodná parametrická vyjádření daných objektů.

#### a) Detekce přímek

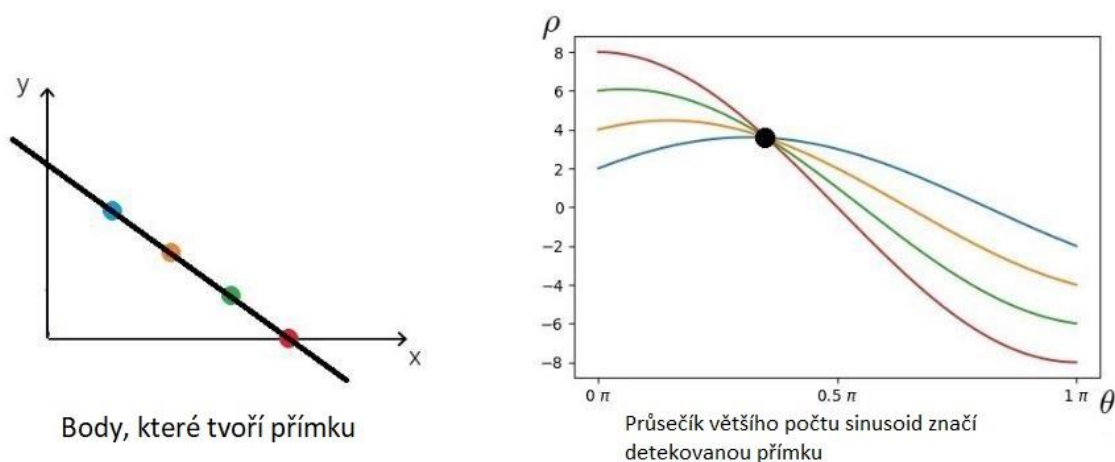
Podstatou detekce přímky je nalezení parametrických rovnic (3) v polárních souřadnicích pro každý pixel, který byl označen jako hrana. Parametr  $\theta$  se mění od 0 do  $\pi$  a dopočítává se  $\rho$ , jejich kombinací se získává sinusoida pro daný bod v  $\theta - \rho$  prostoru. Bod, kde se v tomto prostoru protíná více sinusoid, než stanovuje podmínka, určuje přímku v prostoru obrazu, danou těmito parametry.

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (3)$$

, kde  $\rho$  ... nejkratší vzdálenost od počátku k přímce  
 $\theta$  ... úhel mezi osou x a přímkou  
 $x, y$  ... souřadnice hrany



Obr. 21: Získání sinusoidy z bodu [24]



Obr. 22: Určení přímky průsečíkem sinusoid [24]

#### b) Detekce kružnice

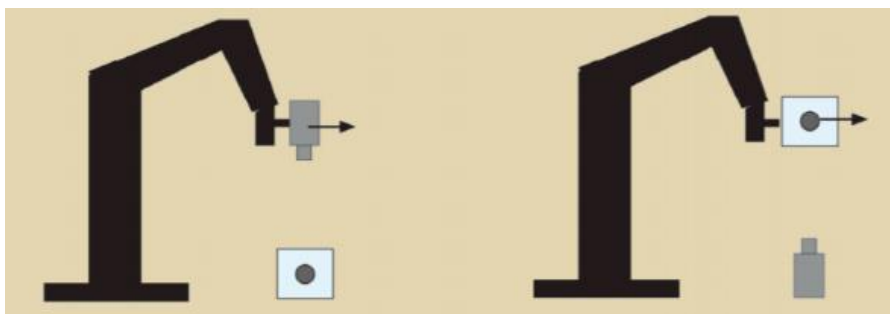
Detekce kružnic pracuje na podobném principu jako detekce přímek, avšak zde je objekt popsán třemi parametry. Vychází to z parametrického vyjádření kružnice (4). Tentokrát se tedy hrany vyjadřují v trojdimenzionálním prostoru daném  $r, x_0, y_0$  a hledají se jejich průsečíky. Třetí rozměr značně zvyšuje výpočetní náročnost, proto se používají metody využívající gradientu.

$$r^2 = (x - x_0)^2 + (y - y_0)^2 \quad (4)$$

, kde  $x, y$  ... souřadnice hrany  
 $x_0, y_0$  ... souřadnice středu kružnice  
 $r$  ... poloměr kružnice

### 3.6. Visual servoing

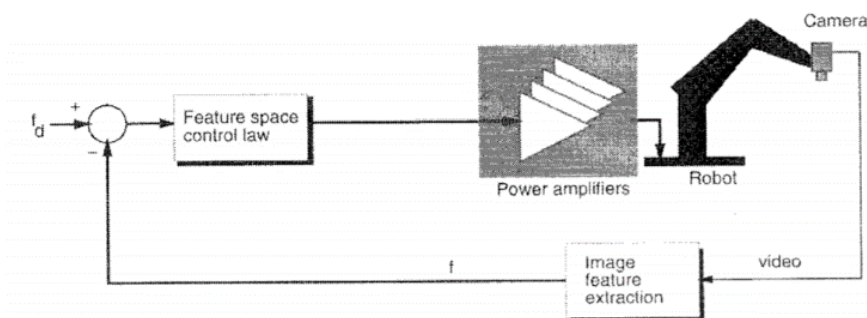
Visual servoing je metoda, která využívá informace získané z kamery za účelem ovládání pohybů robota nebo jeho částí. Existují dva základní přístupy. První z nich je eye-in-hand, což znamená, že senzor je umístěn na robotovi a snímá jeho relativní pozici v prostoru. Eye-to-world znamená, že je senzor připevněn v prostoru a snímá pohyb robota odtud.



Obr. 23: Visual servoing – eye-in-hand vlevo, eye-to-hand vpravo [25]

#### a) Image based visual servoing

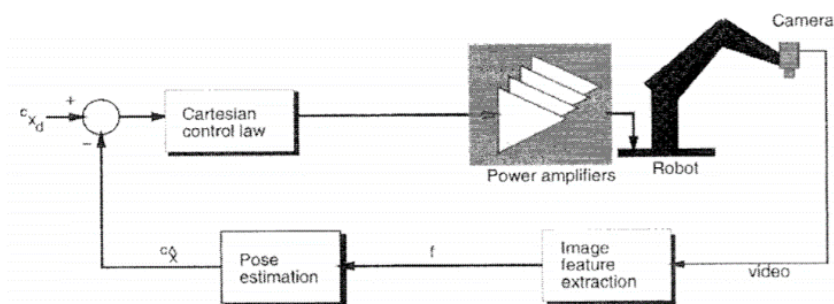
Tento přístup detekuje identifikační body a jejich polohu v snímku z kamery. Zároveň má stanovenou požadovanou polohu těchto bodů. Porovnává tedy aktuální snímek s tím, jak by měl snímek vypadat, aby byl robot ve správné pozici. Z rozdílu snímků vypočítá, jakým způsobem se má robot pohybovat, aby dosáhl požadovaného bodu.



Obr. 24: Image based visual servoing [26]

#### b) Position based visual servoing

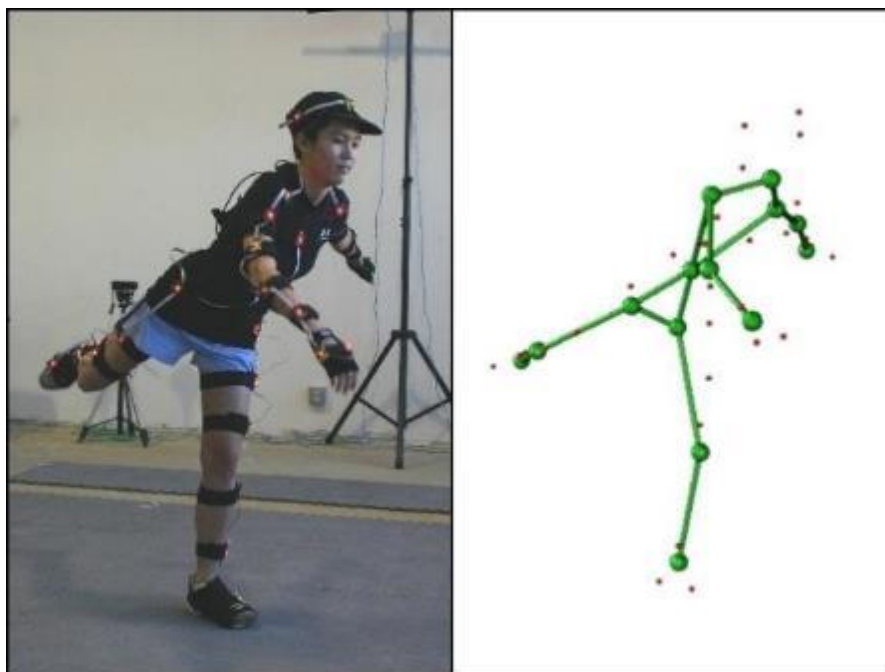
Tento přístup detekuje identifikační body a následně z nich počítá vhodnými geometrickými rovnicemi v kartézských souřadnicích aktuální polohu. Tu porovnává s požadovanou polohou a na základě chyby určuje, jak se má robot pohybovat.



Obr. 25: Position based visual servoing [26]

### 3.7. Motion capture

Motion capture je proces, kterým se získávají informace o pohybu subjektu. Obvykle se využívá v zábavním průmyslu (počítačové hry, filmy), kde se díky němu získávají realistické pohyby herců a ty se graficky upravují pro potřebné účely. Princip spočívá v získání bodů, které slouží pro počítačovou analýzu pohybů. Na subjekt se umístí vhodný počet značek (často koule) na vhodná místa tak, aby bylo možno pohyb jednoznačně zachytit a získat z něj informace – rychlost, úhel, vzdálenost.



*Obr. 26: Motion capture – příklad využití [27]*

### 3.8. Tkinter

Posledním bodem zadání je tvorba demonstrační úlohy, ale vzhledem k vyhlášenému nouzovému stavu hrozilo, že na tento úkol nebude dostatek času. Alternativně tedy bylo dohodnuto, že se místo toho vytvoří grafické uživatelské rozhraní, kterým bude možno robota ovládat. Grafické uživatelské rozhraní (GUI) je rozhraní, díky kterému lze ovládat programy pomocí grafických ovládacích prvků. Python nabízí vícero knihoven, které lze k tomuto účelu využít, aktuálně nejoblíbenější je Tkinter, který je dodáván již s instalací Pythonu. Je populární i díky jednoduchosti, kvůli tomu se ale hodí spíše na projekty menšího rozsahu. Existuje spousta tutoriálů, během kterých se dá s Tkinterem naučit pracovat. Z těchto důvodů bylo rozhodnuto, že GUI bude psáno právě v něm.

V Tkinteru se funkcionality implementují pomocí widgetů, což je programovatelný prvek, který vykonává požadovanou akci. Je zde widget na tlačítko, na menu, na zaškrtnuté políčko a spoustu dalších. Widgety se poté umísťují do okna jedním ze tří geometry managerů – place, pack, grid. Spuštěný program běží cyklicky a obsluhuje jednotlivé akce na základě vstupů uživatele.

## 4. Mechanická konstrukce

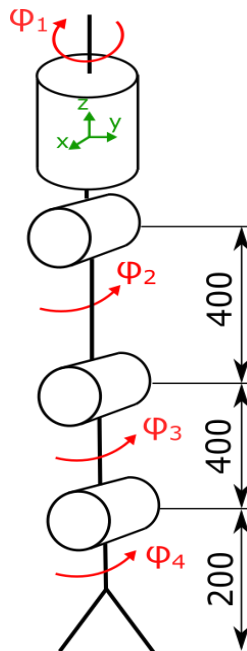
Tato sekce se ve své první části zabývá konstrukcí ruky se čtyřmi stupni volnosti, inspirující se paletovými manipulátory. Proces se skládá z několika dílčích úkolů, mezi které patří výběr pohonů, řešení konstrukčních uzlů, návrh efektoru a na závěr shrnutí celého systému. Následně je uveden stručný popis druhé ruky, kterou má ve své práci podrobně rozebranou kolega [1]. Dalším krokem je vytvoření základny, na kterou se obě ruce uloží. Ta bude konstruována tak, aby byl umožněn její vertikální pohyb.

Pro účely stručnějšího značení a identifikace bylo stanoveno značení jednotlivých vazeb, a to následovně: pro ruku se čtyřmi stupni volnosti se používá římská I, pro druhou ruku římská II. Může se vyskytnout i označení levá pro jednodušší a pravá pro složitější paži. Pohony jsou číslovány 1-4, respektive 1-6, přičemž 1 je nejbližší základně a 4 nebo 6 nejbližší efektoru. Efektor je značen písmenem E. Výsledný tvar pro čtvrtý motor na ruce se šesti stupni volnosti je tudíž II4. Zvláštní označení H nese lineární posuv, podle heveru.

### 4.1. Robotická paže 4 DoF

#### 4.1.1. Návrh řešení

Konstrukce respektuje podobu paletových manipulátorů, oproti vzorovým průmyslovým řešením byla modifikována poloha první vazby – u průmyslových manipulátorů je uložení prvního motoru nejnižší částí robota, zde je tomu naopak. Tato modifikace byla provedena mimo jiné kvůli tomu, aby ruka robota svým pohybem připomínala lidskou ruku. V rámci návrhu byla určena přibližná vzdálenost vazeb tak, aby byl pracovní prostor dostatečně velký. Výsledné řešení se může lišit v závislosti na dostupných materiálech. Zakončení poslední vazby bude obsahovat efektor, jehož design závisí na požadovaných funkcích. Jelikož účelem manipulátoru jsou zejména demonstrační úkony, měl by efektor umět uchopit například dřevěnou kostku či míček.



Obr. 27: Kinematické schéma návrhu řešení

#### 4.1.2. Volba pohonů

Klíčovým aspektem výběru pohonů se stala nízká pořizovací cena, konstrukce je tedy značně uzpůsobená dostupným motorům. Bylo sehnáno několik elektromotorů z automobilů, a to ze stěračů (předních i zadních) a mechanismů pro zvedání oken. Všechny pohony jsou dimenzovány na 12 V a jsou vybaveny šnekovou převodovkou, což má pro robotickou ruku jednu významnou výhodu, a tou je samosvornost.

Motory z předních stěračů (obr. 29 a 32) mají neomezený rozsah pohybu, proto najdou uplatnění tam, kde je žádoucí velký rozsah rotace. Nevýhodou je větší hmotnost a rozměry, nehodí se proto na místa, kde by způsobovaly příliš velký zátěžný moment.

Motory ze zadních stěračů (obr. 33) jsou oproti předním stěračům kompaktnější a lehčí. Jejich nevýhodou je omezený rozsah natočení cca 110°. Při dosažení krajní polohy se změní směr rotace, je proto nutno zajistit, aby se krajové polohy nedosáhlo.

Mechanismus pro elektrické ovládání oken (obr. 31) se skládá ze dvou částí. První z nich je motor se šnekovou převodovkou, na jejíž výstupní hřídeli je pastorek, který pohání druhou část mechanismu. Ten se skládá z ozubení na pásovině, která je spojena s dalšími díly. Důsledkem dvou převodovek je vysoký moment a také nejjednodušší regulace a minimální vůle. Rozsah pohybu mechanismu je omezen cca na 100°.

Z dostupných pohonů byl vybrán pro každou vazbu ten nejvhodnější při zohlednění požadavků, jejichž soupis je dán těmito body:

1. Rozměry
2. Hmotnost
3. Možnosti uložení
4. Vůle
5. Výstupní moment

Byly sestaveny možné kombinace umístění pohonů a provedl se test, zda jsou motory schopny generovat dostatečný moment. Na výstupní hřídel se připevnilo rameno o délce představující očekávanou vzdálenost těžiště od osy rotace. Na konec ramena bylo zavěšeno závaží o hmotnosti, která reprezentovala předpokládanou hmotnost ramene. Motor se pevně ukotvil a po přivedení napětí se musel bez problémů hýbat. Finálně bylo rozhodnuto, že jednotlivé vazby (dle obr. 32) budou poháněny následujícími motory:

- $\varphi_1$ : Motor z předního stěrače (obr. 29)
  - Vysoká hmotnost a nekompaktní rozměry – nevadí, umístěn na těle
  - Nižší moment – nevadí, překonává pouze tření a setrvačnost
- $\varphi_2$ : Mechanismus zvedání oken (obr. 31)
  - Vysoký moment
  - Potřebuje nejvíce prostoru – nevadí, toto rameno je plánováno nejdelší
- $\varphi_3$ : Motor z předního stěrače (obr. 32)
  - Kompromis hmotnosti a momentu
  - Vyžadován velký rozsah pohybu
  - Vyšší vůle
- $\varphi_4$ : Motor ze zadního stěrače (obr. 33)
  - Nízká hmotnost
  - Vzhledem ke konstrukci nevadí omezený rozsah pohybu
  - Malé momentové nároky

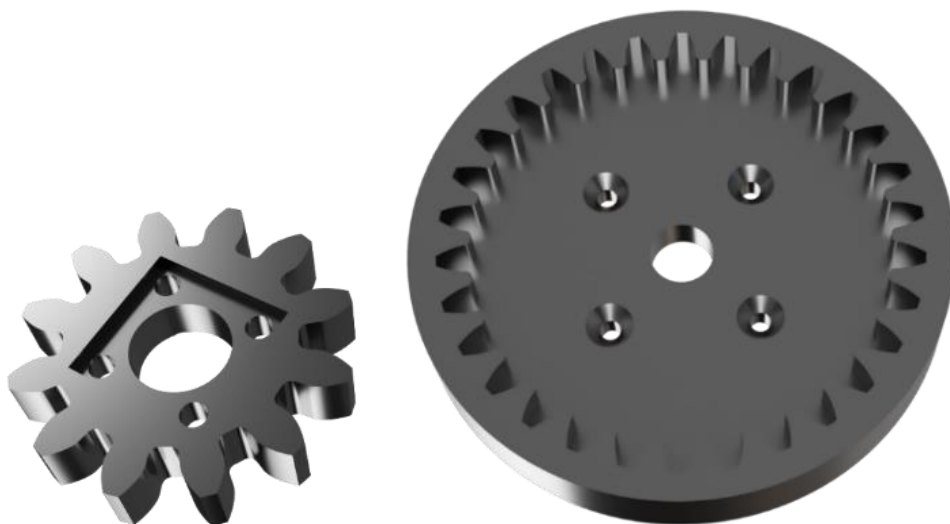


### 4.1.3. Konstrukce

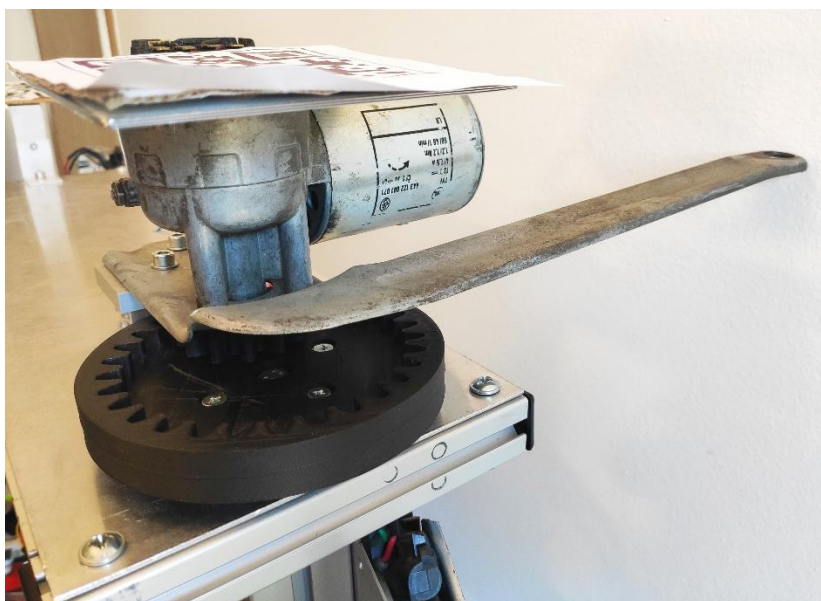
V této kapitole jsou rozebrány důležité konstrukční uzly a jejich řešení. První rotace probíhá ve vertikálním směru, zbylé tři v horizontálním. Většina konstrukce je složena z ocelové pásoviny o průřezu 30x2 mm a plastových dílů z tiskárny. Součástí jsou i výztuhy instalované kvůli snížení kmitání a torzních průhybů. Pro realizaci vazeb se často využívala ložiska ze starých kolečkových bruslí. Mají vnitřní průměr 8 mm a vnější průměr 22 mm, ukládají se do navrhnutých ložiskových domků. Hřídel s pojistnými kroužky je nahrazena M8 šroubem, který je připevněn ke konstrukci různými způsoby.

#### a) První vazba $\varphi_1$

Tato vazba je umístěna v těle a její součástí je převodovka, jejímž cílem je zjemnit chod a snížit vůli. Pastorek má 12 a vnější ozubené kolo 30 zubů, převodový poměr je  $i = 2,5:1$ . Modul ozubení je 4.

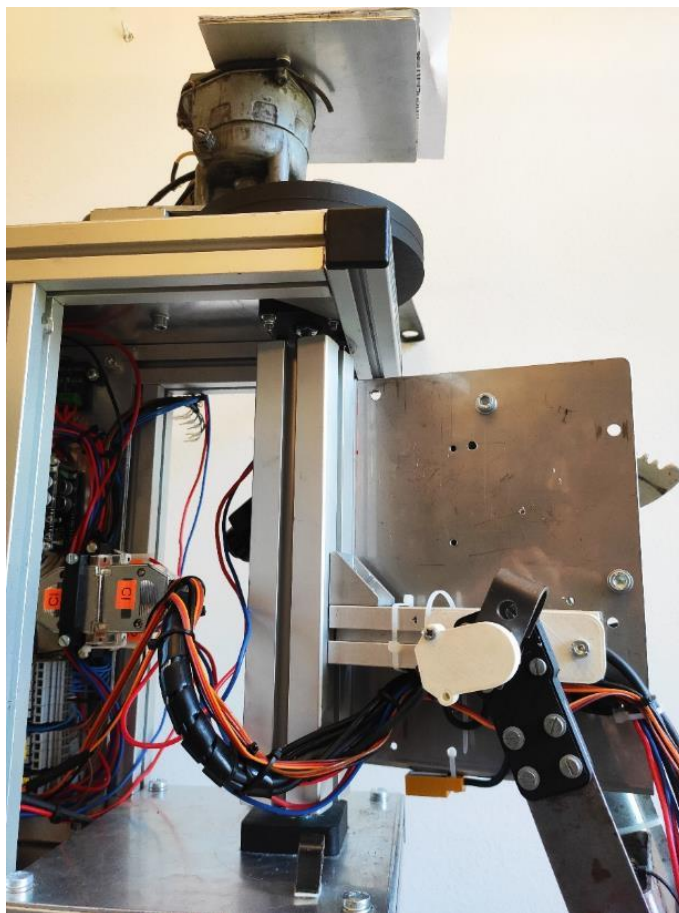


Obr. 28: Převodovka – pastorek a ozubené kolo



Obr. 29: První vazba – pohled shora

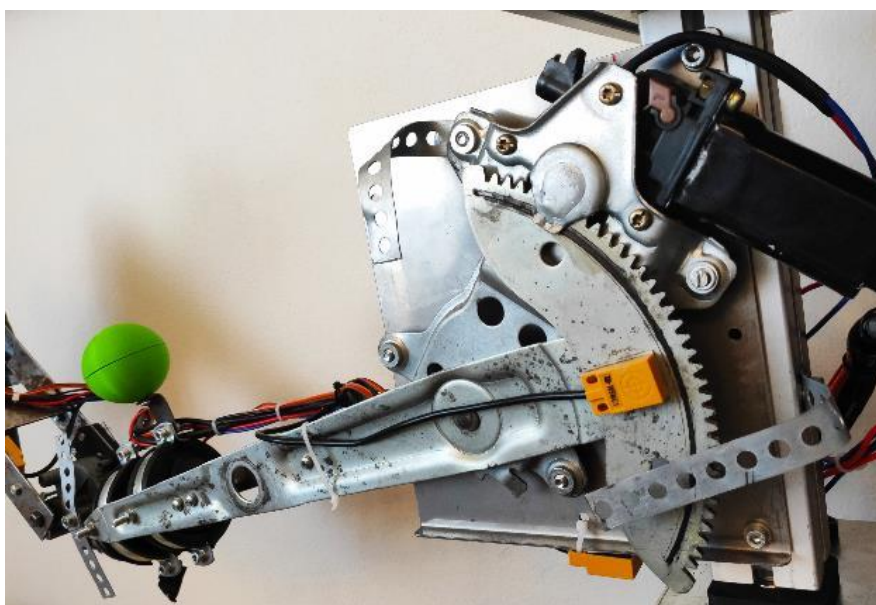




*Obr. 30: První vazba*

b) Druhá vazba  $\varphi_2$

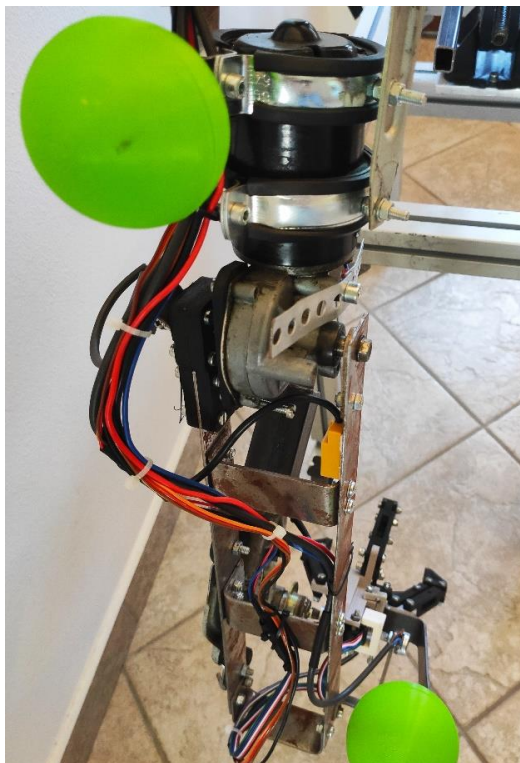
Jako druhý pohon slouží mechanismus elektrického ovládání okének, který tvoří jedno rameno. Druhé je tvořeno pásovinou, v horní části připevněnou k ložiskovému domku. Na konci se ramena spojí dvoušroubovou objímkou s gumovým vnitřkem.



*Obr. 31: Druhá vazba*

c) Třetí vazba  $\varphi_3$

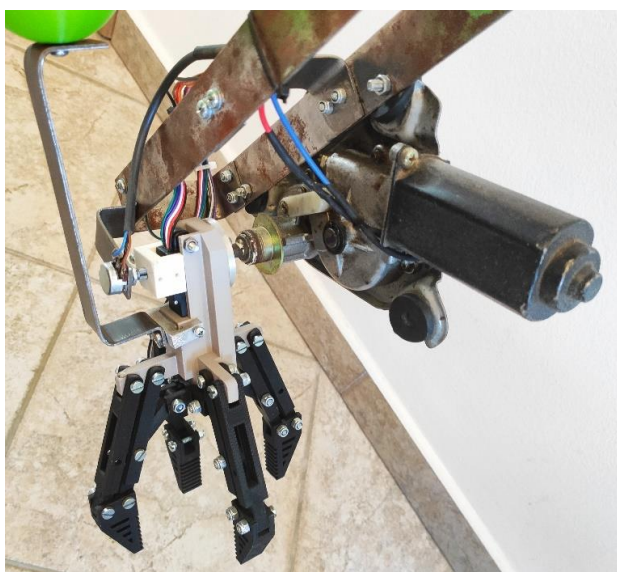
Výše zmíněné dvě objímky slouží k uložení třetího motoru. Rameno ve tvaru U je připevněno k výstupní hřídeli motoru a na druhé straně k ložiskovému domku. Tím prochází šroub, upevněný ke krytu převodovky.



*Obr. 32: Třetí vazba*

d) Čtvrtá vazba  $\varphi_4$

Poslední pohon je přišroubován k předchozímu rameni na pravé straně. Výstupní hřídel slouží k instalaci kompletní sestavy efektoru přes hliníkový díl, který se s hřídelí spojí červíkem.



*Obr. 33: Čtvrtá vazba*

#### 4.1.4. Efektor

Aby ruka mohla manipulovat s předměty, musí být zakončena funkčním efektem. Před návrhem byl nalezen nevyužívaný lineární motor Actuonix L12-30-210-12. Jelikož je lehký, malý a má velkou sílu, efektor byl navrhnut tak, aby jím byl poháněn.

Veličina	Hodnota
Hmotnost	34 g
Napájecí napětí	12 V
Maximální výsuv	30 mm
Převodový poměr	210:1
Maximální rychlost	6,5 mm/s
Maximální síla	80 N

Tab. 5: Parametry lineárního pohonu [28]



Obr. 34: Lineární motor Actuonix L12 [28]

Inspiraci pro návrh tohoto efektoru poskytlo řešení na obr. 40, to se vyznačuje úchopy, které při pohybu mají stále stejný sklon vůči vertikální ose efektoru. Toho je docíleno upevněním těchto úchopů na dvě rovnoběžná ramena o shodné délce. Oproti vzoru má efektor čtyři ramena, aby šlo snáze uchopovat symetrické předměty.



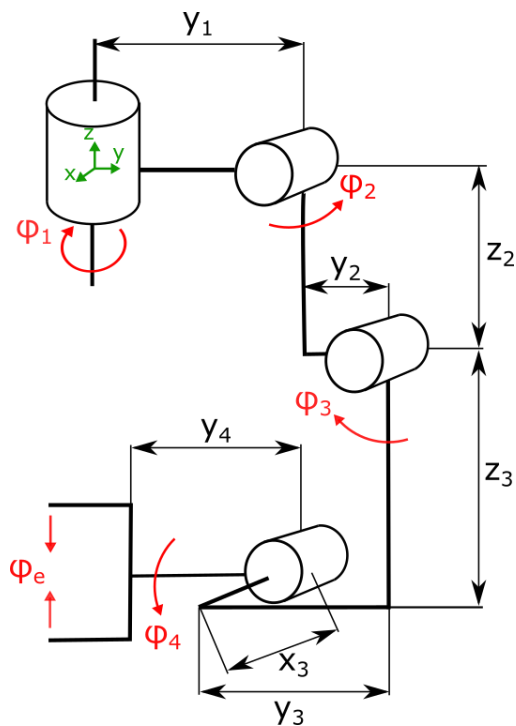
Obr. 35: Robotické chapadlo Makeblock [29]



Obr. 36: Model efektoru v prostředí Fusion 360

Efektor se skládá z 18 dílů. Základna byla vymodelována tak, aby se mohla přišroubovat k hliníkovému dílu, připevněnému na výstupní hřídel motoru. Do středu kříže uprostřed je umístěno zakončení lineárního motoru, v důsledku jeho výsuvu dochází pákovým mechanismem k otevírání a zavírání efektoru. Při maximálním vysunutí motoru je vzdálenost mezi protilehlými gripy 120 mm. Při plném zavření 40 mm. Sestava má na výšku maximálně 196 mm. Šířka se pohybuje mezi 90 mm a 168 mm.

#### 4.1.5. Shrnutí



Obr. 37: Kinematické schéma I

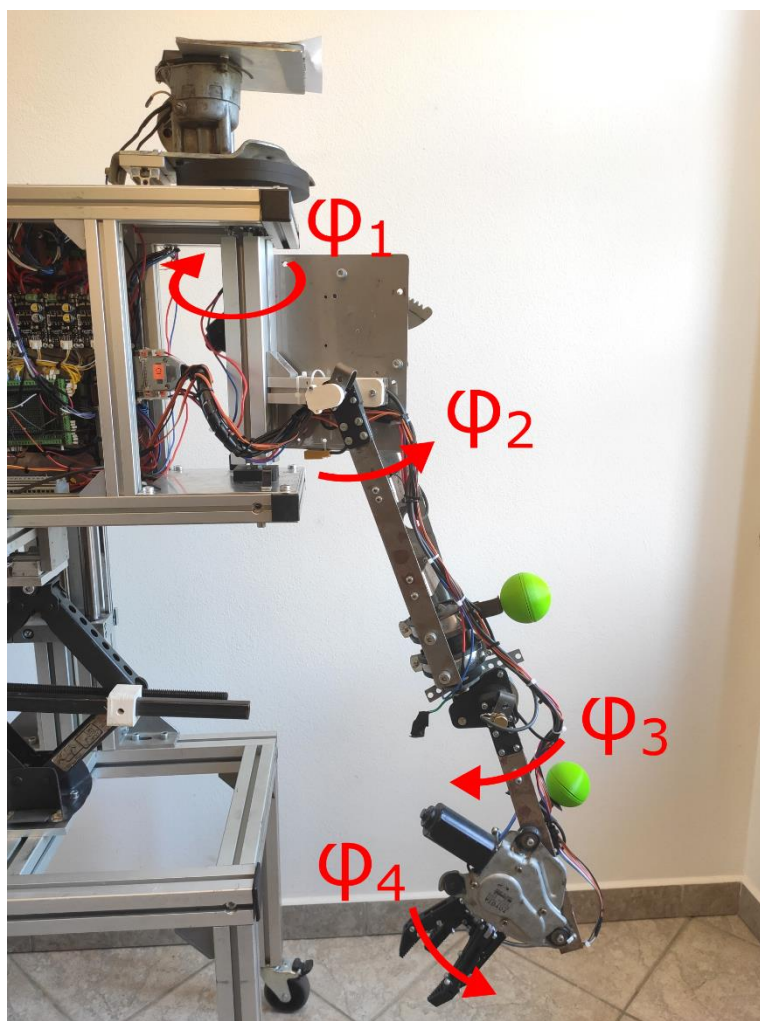


Parametr	Rozměr [mm]
$y_1$	124
$y_2$	35
$z_2$	390
$x_3$	35
$y_3$	50
$z_3$	280
$y_4$	150

Tab. 6: Parametry kinematiky

Vazba	Minimum	Maximum
$\varphi_1$	-1,7 rad	1,7 rad
$\varphi_2$	-0,1 rad	1,6 rad
$\varphi_3$	-2 rad	2 rad
$\varphi_4$	-0,8 rad	0,8 rad
$\varphi_e$	30 mm	120 mm

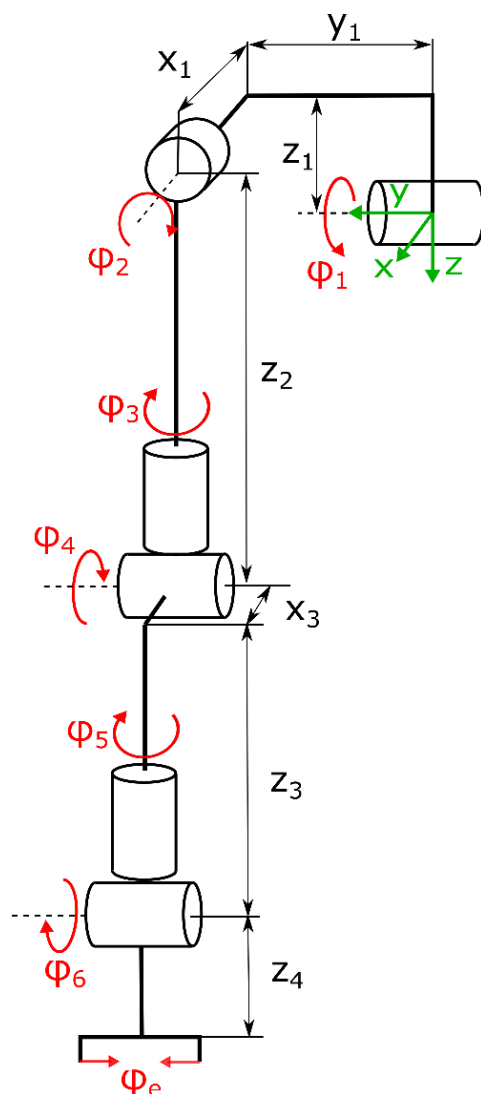
Tab. 7: Fyzické limity vazeb



Obr. 38: Kompletní manipulátor I

## 4.2. Robotická paže 6 DoF

V této krátké kapitole bude informativně popsána druhá ruka, kterou měl na starost Bc. Jakub Chlaň [1]. Důvodem je, že i tato práce ji dále využívá – vybírá pro ni řídicí a výkonovou elektroniku, zpracovává na ní vizuální zpětnovazební řízení. Ruka má šest stupňů volnosti, jsou rozloženy tak, aby přibližně kopírovaly kinematiku lidské paže.



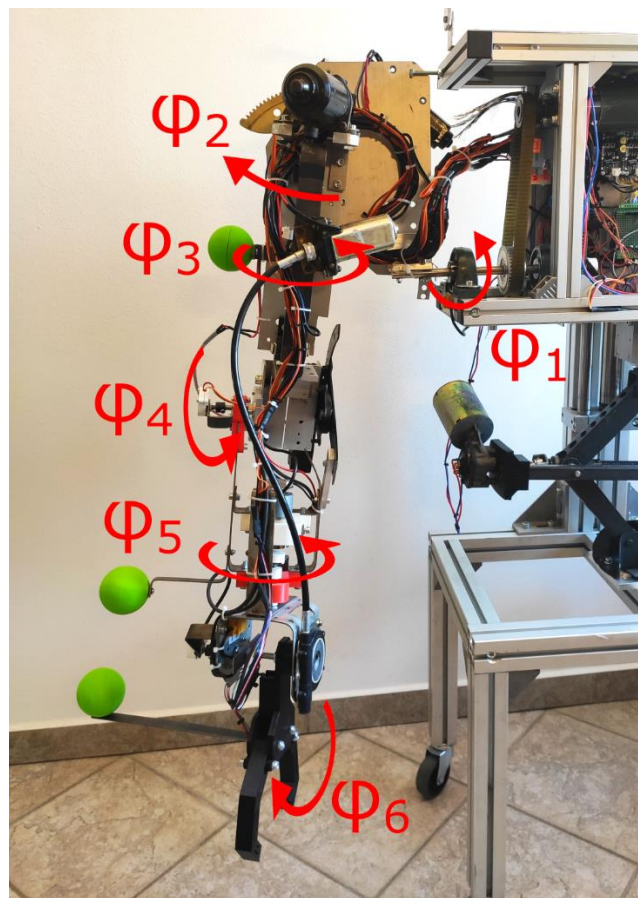
Obr. 39: Kinematické schéma II

Parametr	Rozměr [mm]
$x_1$	35
$y_1$	179
$z_1$	137
$z_2$	310
$x_3$	65
$z_3$	300
$z_4$	220

Tab. 8: Parametry kinematiky II

Vazba	Minimum	Maximum
$\varphi_1$	$-\pi \text{ rad}$	$\pi \text{ rad}$
$\varphi_2$	$-0,1 \text{ rad}$	$1,6 \text{ rad}$
$\varphi_3$	$-\frac{\pi}{2} \text{ rad}$	$\frac{\pi}{2} \text{ rad}$
$\varphi_4$	$-0,1 \text{ rad}$	$1,6 \text{ rad}$
$\varphi_5$	$-1,7 \text{ rad}$	$1,3 \text{ rad}$
$\varphi_6$	$-1,8 \text{ rad}$	$1,8 \text{ rad}$
$\varphi_e$	$10 \text{ mm}$	$100 \text{ mm}$

Tab. 9: Fyzické limity vazeb II



Obr. 40: Kompletní manipulátor II

#### 4.3. Základna a tělo

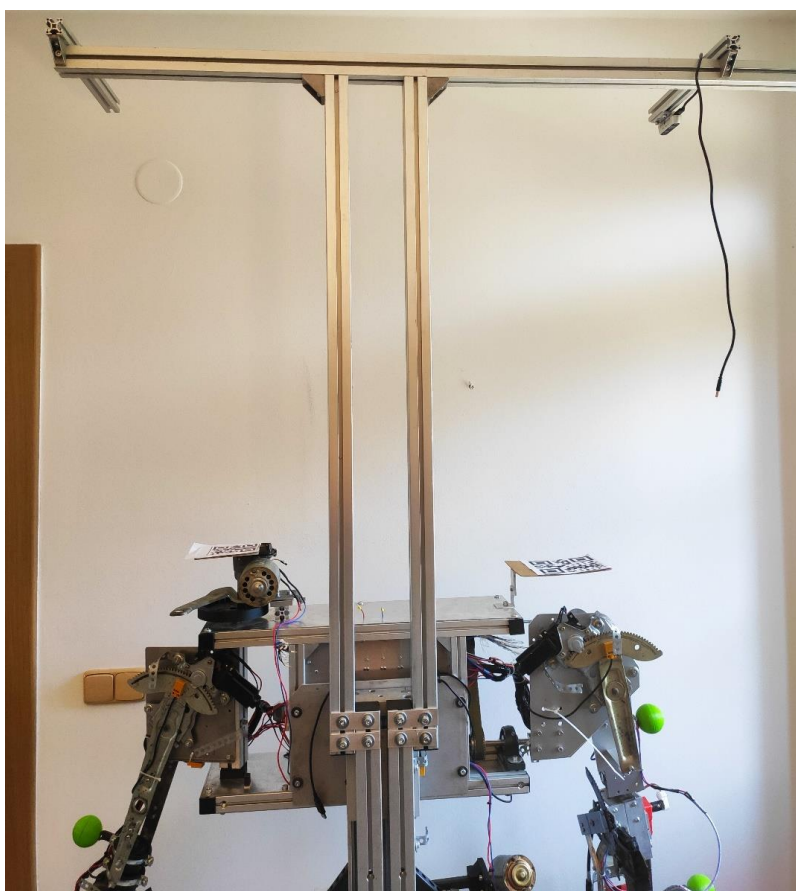
Pro uložení končetin byla postavena konstrukce, kterou tvoří základna a na ní umístěné tělo s rukama. Konstrukce se skládá z hliníkových profilů průřezů 30 a 45 mm. Bylo upřednostněno využití hliníkových profilů, protože stavebnicový systém umožňuje snadnou manipulaci, poskytuje dobrý kompromis mezi váhou a pevností a umožňuje jednoduchou instalaci různých modulů. Nové hliníkové profily a příslušenství byly příliš drahé, avšak podařilo se sehnat použité z obchodu, který se zabývá prodejem dílů z vysloužilých výrobních linek.

#### 4.3.1. Základna

Základna se skládá z profilů o tloušťce 45 mm, je obdélníková a má čtyři nohy s kolečky, z nichž dvě jsou vybavena brzdou. Vzadu jsou dva vertikální profily, ke kterým je připevněn lineární pojezd. Dále se ve středu základny nachází další profil, jehož účelem je zpevnění konstrukce a instalace heveru. Vrchní částí základny je konstrukce ve tvaru písmene T, jejím účelem je nést kameru, využívanou pro vizuální zpětnou vazbu.



*Obr. 41: Základna robota*



*Obr. 42: T konstrukce na kameru*

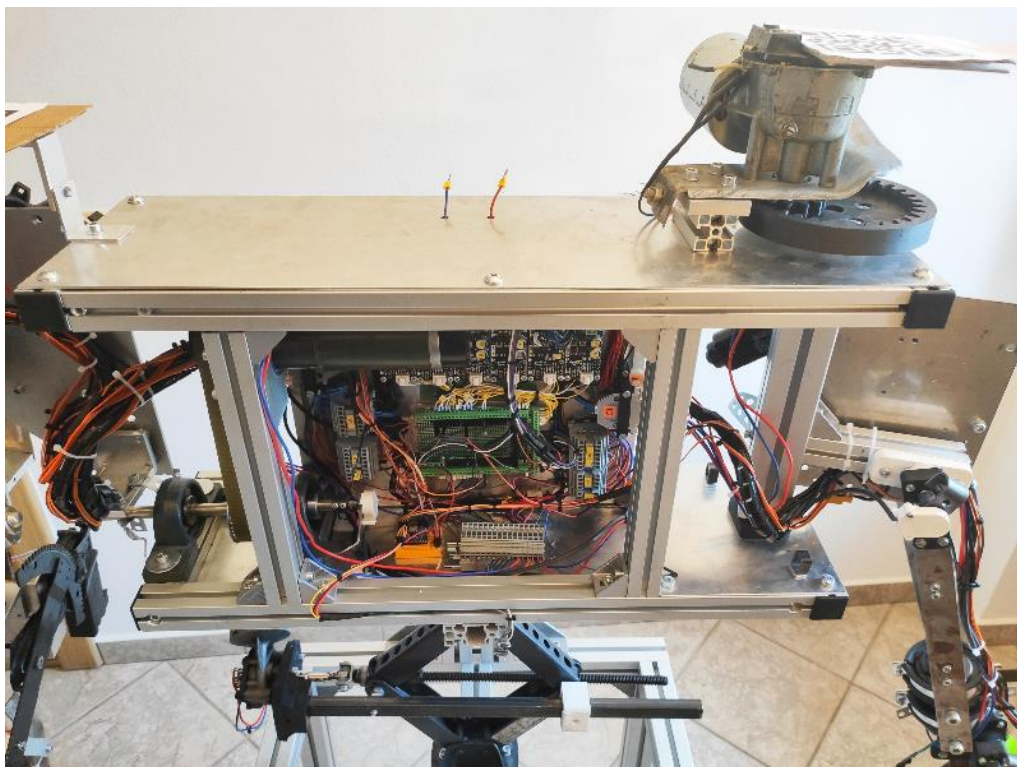


#### 4.3.2. Tělo

Při úvahách nad konstrukcí těla bylo zformulováno několik bodů, které by mělo tělo splňovat, aby bylo schopnost plnit všechny potřebné funkce:

1. Nést dvě paže.
2. Umožnit budoucí montáž hlavy.
3. Mít v sobě rozvaděč (výkonová, řídicí elektronika, senzorika).
4. Být připevněno k vertikálnímu posuvu.
5. Umožnit instalaci krytů.

Nejjednodušším možným tvarem, splňujícím všechny požadavky, je prostý kvádr. Místo na rozvaděč je vevnitř na zadní stěně. Paže se připevní po stranách. Tělo je složeno z 30 mm profilů a je částečně zakrytováno plechy, do budoucna je možno zakrytovat i zbytek průhledným plexisklem.



*Obr. 43: Tělo robota*

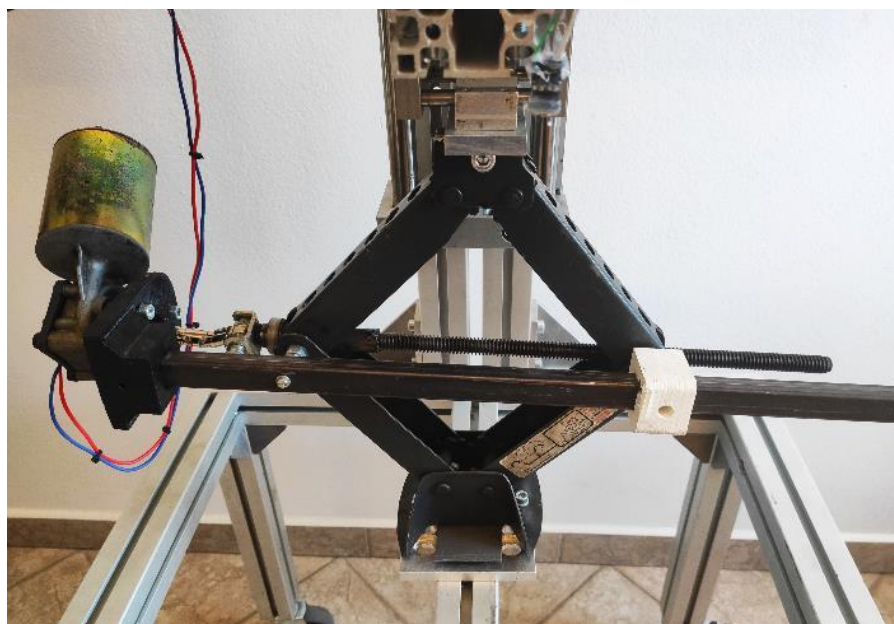
#### 4.3.3. Vertikální posuv

Základna je s tělem spojena ve dvou bodech. Jedním z nich je improvizované lineární vedení, které vzniklo využitím vyřazeného pneumatického válce Festo SLM-32-170-KG a které je připevněno k základně statickou částí a k tělu pohyblivou částí. Fyzická konstrukce omezuje vertikální rozsah pohybu na cca 20 cm.



*Obr. 44: Uložení lineárního vedení*

Vedení pouze určuje směr pohybu, další část posuvu tedy musí přenášet zátěž a zajišťovat vertikální pohyb celého těla. Cenově dostupnou variantou je nůžkový hever na auta a jelikož má být pohyb realizován bez lidského zásahu, je využit DC motorem ze stěrače, který je se šroubem heveru spojen kardanovým kloubem. V důsledku konstrukce heveru dochází při vysouvání k pohybu do strany. Proto byla vytvořena konstrukce, která se umístila mezi hever a tělo tak, aby se tento pohyb mohl vykonávat, a to díky dvojici ložiskových domků.



*Obr. 45: Elektrický hever*

## 5. Elektronika

Tato kapitola se zabývá podrobnějším popisem vybraných řešení v oblasti výkonové a řídicí elektroniky, dále její instalaci, a to včetně popisu rozvaděče a schémat zapojení.

### 5.1. Výkonová

#### 5.1.1. Požadavky

Kompletní robot obsahuje jedenáct pohonů, je tudíž potřeba sehnat stejný počet H můstků. Výkonové obvody budou napájeny ze zdroje stejnosměrného napětí 12 V. Motory mají různé odpory vinutí a tím odebírají i různé proudy při maximálním napětí. Nejnižší odpor je 1  $\Omega$ , H můstky tedy musí být schopny kontinuálně přenášet až 12 A. Preferovaná bude varianta, kde k řízení stačí jeden PWM signál.

#### 5.1.2. Cytron MD13S

Tento H můstek od firmy Cytron je schopen kontinuálně přenášet proud 13 A, je určen pro napětí mezi 6 a 30 V a má ochranu proti nadproudu i podpětí. Dokáže fungovat na 3,3 i 5 V logiku a zvládá frekvence PWM až do 20 kHz. K jeho řízení jsou zapotřebí 3 piny – 1x GND, 1x PWM, 1x DIR. S pořizovací cenou okolo 250 Kč je ve všech ohledech vhodnou volbou.



Obr. 46: Cytron MD13S [8]

### 5.2. Řídicí

#### 5.2.1. Požadavky

Pro výběr řídicí elektroniky je jedním z klíčových aspektů požadavek na počet pinů. Robot využívá jedenáct pohonů, to znamená jedenáct PWM a digitálních výstupů. Oba efektorů jsou poháněny lineárními pohony, jejichž vysunutí se řídí střídou, to znamená dva PWM výstupy. Další požadavky vyplývají ze sensoriky, která je popsána v práci [1]. Ke snímání aktuální polohy jsou na pohonech na rukách instalovány potenciometry, jejichž výstupem je napětí. Na pohonech, u kterých hrozí nebezpečí při dosažení krajní polohy, jsou instalovány koncové spínače. Těch je celkem devět a sepnutí je snímáno digitálním vstupem. Proudů tekoucí do motorů snímají proudové senzory, které mají jako výstup napětí. Dále jsou instalovány tři ToF senzory, které snímají vzdálenost povrchu. V každém efektoru je jeden a třetí určuje výšku těla nad základnou. Ke komunikaci využívají I<sup>2</sup>C sběrnici.

Druh pinu	Požadovaný počet pinů
PWM výstup	11
Digitální výstup	11
Digitální vstup	9
Analogový vstup	21
I <sup>2</sup> C piny (SCL, SDA)	3

Tab. 10: Požadavky na počet pinů

### 5.2.2. Arduino Mega 2560

Robot je řízen deskou Arduino Mega 2560. Klíčové aspekty, které vedly k této volbě, jsou zejména dostatečný počet pinů a cenová dostupnost. Zároveň byl zakoupen i shield, který zjednodušuje zapojení pinů, protože jsou na něm svorkovnice se šrouby.

Parametr	Hodnota
Mikrokontroler	ATmega2560 (16 MHz)
Provozní napětí	5 V
Digitální vstupy/výstupy	54 (z toho 15 PWM)
Analogové vstupy	16
Flash paměť	256 kB
Rozměry	101x53 mm

Tab. 11: Základní parametry Arduino Mega 2560



Obr. 47: Arduino Mega 2560 [15]

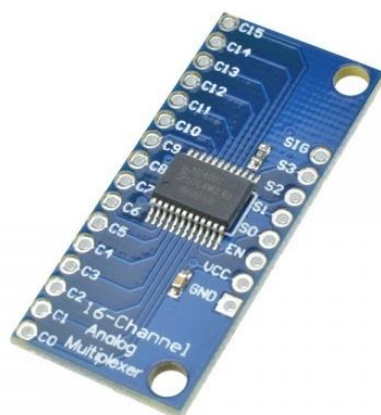
### 5.2.3. Analogový multiplexer CD74HC4067

Arduino nemá pro tuto práci dostatečný počet analogových vstupů, zvýšit jejich počet lze použitím analogového multiplexeru, který v tomto případě nabízí připojení dalších šestnácti analogových vstupů. Zařízení bude využito na odečítání proudů z proudových senzorů.

Na piny C0 – C15 se přivádí analogové vstupy. S Arduinem se multiplexer propojí pomocí sedmi pinů, a to 1x GND, 1x 5 V, 4x digitální výstup z Arduina a 1x signální pin.



Pomocí binární kombinace nul a jedniček na piny S0 až S3 se určí, které napětí se objeví na výstupním signálovém pinu. Tabulka 12 uvádí, v jakém pořadí jsou jednotlivé výstupy z proudových senzorů do multiplexeru připojeny.



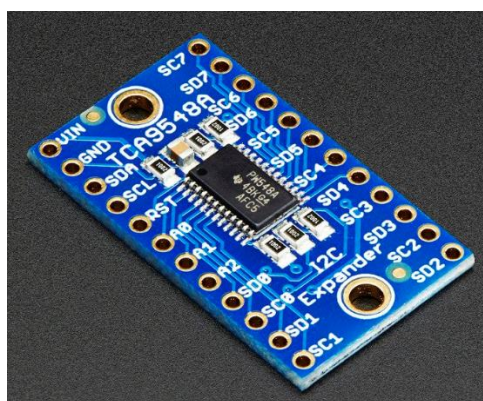
Obr. 48: Analogový multiplexer CD74HC4067 [30]

Pohon	I1	I2	I3	I4	II1	II2	II3	II4	II5	II6	H
Pin	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

Tab. 12: Zapojení proudových senzorů

#### 5.2.4. I<sup>2</sup>C multiplexer TCA9548A

I<sup>2</sup>C multiplexer je zapotřebí, pokud se požaduje zapojení více zařízení se stejnou I<sup>2</sup>C adresou. Na konstrukci jsou umístěny tři Time of Flight senzory vzdálenosti, které komunikují přes tuto sběrnici. Multiplexer TCA9548A umožňuje propojení až osmi zařízení. Výběr konkrétního připojeného zařízení se provádí zasláním bytu s odpovídající hodnotou a až do další změny je komunikováno s takto vybraným zařízením. V tabulce 13 je uvedeno, ke kterým pinům jsou jednotlivé senzory vzdálenosti připojeny.

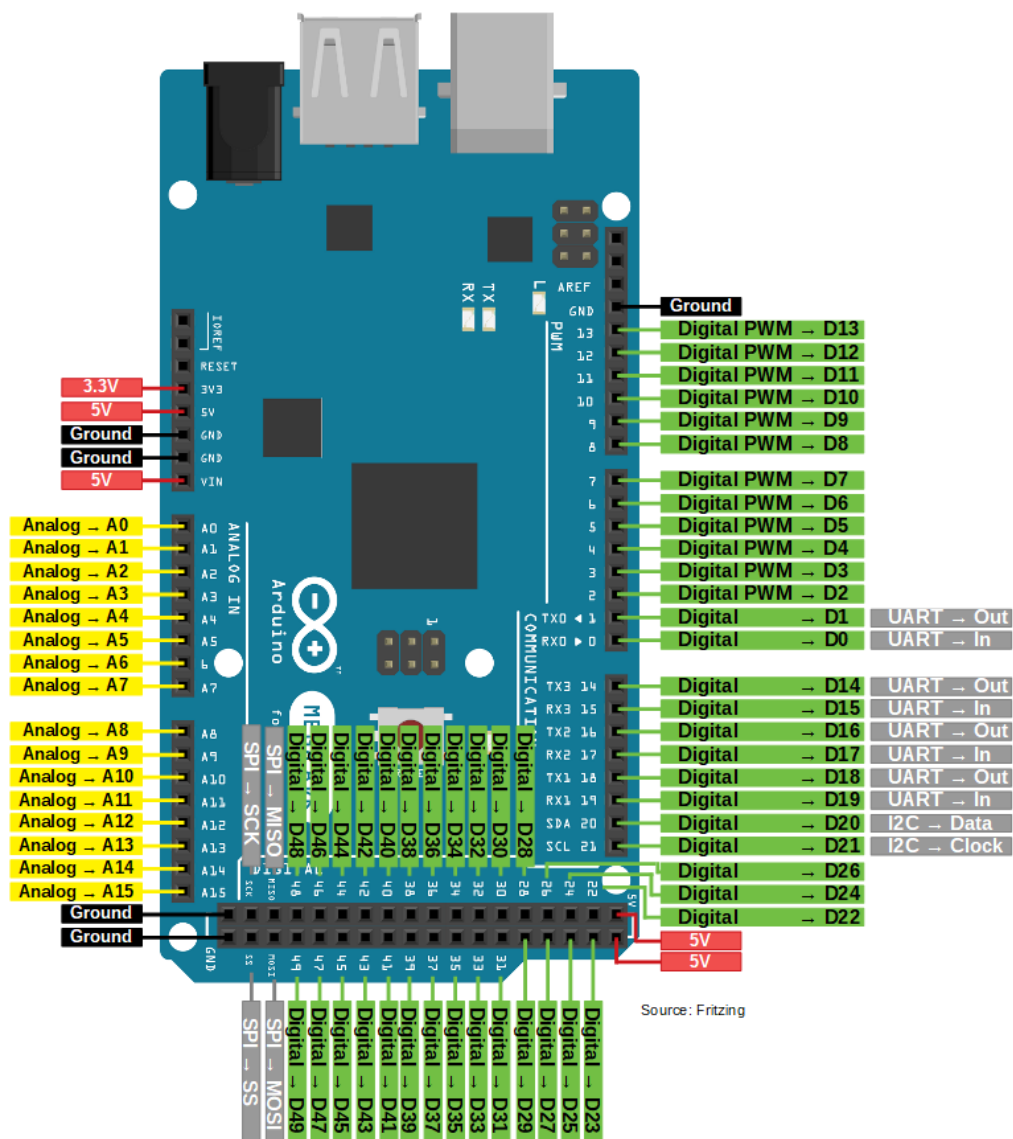


Obr. 49: TCA9548A [31]

ToF senzor	Ruka I	Ruka II	H
Piny	SC1/SD1	SC0/SD0	SC2/SD2

Tab. 13: Zapojení Time of Flight senzorů

### 5.2.5. Pinout Arduino



Obr. 50: Arduino Mega 2560 - pinout [32]

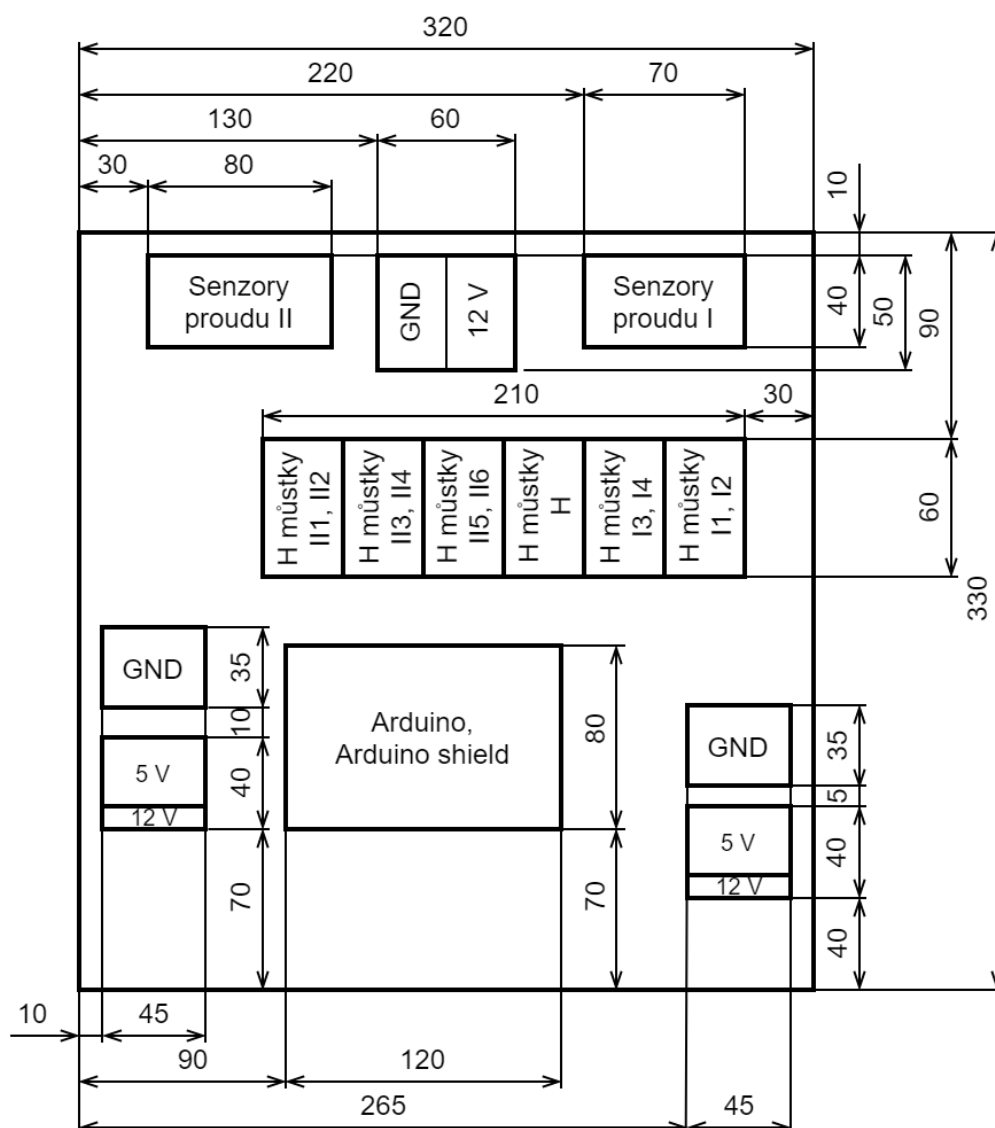
Typ pinu	Číslo pinu	Účel
Analogový vstup	A0	Potenciometr I1
	A1	Potenciometr I2
	A2	Potenciometr I3
	A3	Potenciometr I4
	A4	Potenciometr II1
	A5	Potenciometr II2
	A6	Potenciometr II3
	A7	Potenciometr II4
	A8	Potenciometr II5
	A9	Potenciometr II6
	A10	Analogový multiplexer – signál

PWM výstup	D2	PWM H můstek I1
	D3	PWM H můstek I2
	D4	PWM H můstek I3
	D5	PWM H můstek I4
	D6	PWM IE
	D7	PWM H můstek II1
	D8	PWM H můstek II2
	D9	PWM H můstek II3
	D10	PWM H můstek II4
	D11	PWM H můstek II5
	D12	PWM H můstek II6
	D13	PWM IIE
Digitální výstupy	D22	DIR H můstek I1
	D23	DIR H můstek I2
	D24	DIR H můstek I3
	D25	DIR H můstek I4
	D26	DIR H můstek II1
	D27	DIR H můstek II2
	D28	DIR H můstek II3
	D29	DIR H můstek II4
	D30	DIR H můstek II5
	D31	DIR H můstek II6
	D14	PWM H můstek H
	D15	DIR H můstek H
	D34	Analogový multiplexer – S0
	D35	Analogový multiplexer – S1
	D36	Analogový multiplexer – S2
	D37	Analogový multiplexer – S3
Digitální vstupy	D40	Koncový spínač I1
	D41	Koncový spínač I2
	D42	Koncový spínač I3
	D43	Koncový spínač II1
	D44	Koncový spínač II2
	D45	Koncový spínač II3
	D46	Koncový spínač II5
	D47	Koncový spínač II6
	D48	Koncový spínač H
I2C komunikace	SCL	I <sub>2</sub> C multiplexer – SCL
	SDA	I <sub>2</sub> C multiplexer – SDA

Tab. 14: Zapojení pinů do Arduina

### 5.3. Rozvaděč

Uvnitř těla robota je instalován rozvaděč, na kterém se nachází veškerá potřebná elektronika. Jeho středem je v předešlých podkapitolách popsán Arduino Mega 2560. Okolo něj se nachází dvanáct H můstků, jedenáct proudových senzorů a větší množství svorek na DIN lištu Wago (se dvěma nebo třemi kontakty). Na krajích jsou díly, které slouží k upevnění konektorů na odpojení celé ruky. Napětí ze zdroje je přivedeno na 12 V a GND svorky v horní části rozvaděče. Odtud je rozvedeno do všech H můstků. Kladný pól výstupu z H můstků vede do odpovídajícího proudového senzoru a odtud do konektoru, který spojuje ruku s rozvaděčem. Záporný pól vede přímo do konektoru.



Obr. 51: Schéma rozvaděče

Mimo rozvaděčovou desku jsou instalovány oba multiplexery. Analogový je připevněn k hornímu plechu, I<sup>2</sup>C k plechu spodnímu. Na spodním plechu je ještě několik svorek na DIN lištu, slouží k připojení kabelů z prvních vazeb na obou rukách a heveru.



## 6. Zpracování obrazu

### 6.1. Intel Realsense D435

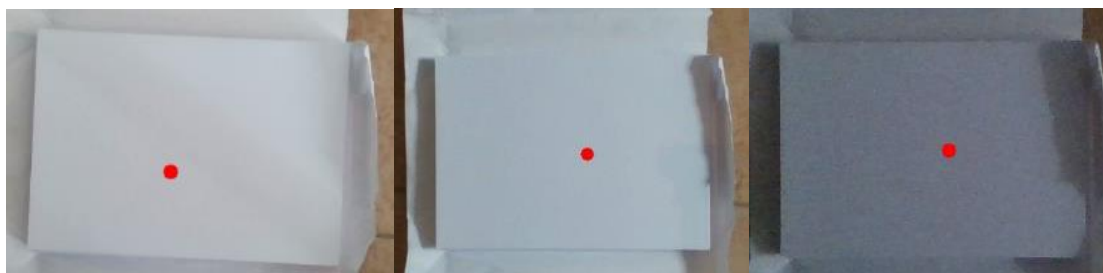
Pro účely vizuálního zpětnovazebního řízení byl zakoupen senzor Intel Realsense D435, a to zejména z těchto důvodů:

1. Široký zorný úhel  $86^\circ$  – umožňuje umístění senzoru blíže konstrukci
2. Kompaktní rozměry – 90 x 25 x 25 mm
3. Vysoké rozlišení – hloubka 1280 x 720, 90 FPS; RGB 1920 x 1080, 30 FPS
4. Vlastní software, vzorové kódy
5. Dostupnost v České republice

#### 6.1.1. Otestování vybraných funkcionalit

##### a) Přesnost hloubkového senzoru

Přesnost hloubkového senzoru byla otestována porovnáním přesně změřené vzdálenosti s daty z kamery. Testované hloubky reprezentovaly rozsah vzdáleností, ve kterých budou snímány robotické paže – bližší značka se umístila 800 mm od kamery, vzdálenější 2000 mm od ní. Následně se sejmulo 500 vzorků, z nichž se vypočítala průměrná hodnota. Testy se provedly za denního světla, při umělém osvětlení a za šera.



Obr. 52: Snímání plochy vzdálené 800 mm – denní světlo, umělé osvětlení a šero

<i>Podmínky</i>	<i>Skutečná vzdálenost [mm]</i>	<i>Průměr ze senzoru [mm]</i>	<i>Rozdíl [mm]</i>	<i>Rozdíl [%]</i>
<i>Denní světlo</i>	800	794	-6	0,75
<i>Umělé osvětlení</i>	800	797	-3	0,38
<i>Šero</i>	800	799	-1	0,13
<i>Denní světlo</i>	2000	1954	-46	2,30
<i>Umělé osvětlení</i>	2000	1919	-81	4,05
<i>Šero</i>	2000	1932	-68	3,40

Tab. 15: Srovnání skutečné a měřené vzdálenosti objektu od senzoru

Z výsledků je patrné, že přesnost s rostoucí vzdáleností klesá. Pro plochu vzdálenou 800 mm lze chybu považovat za zanedbatelnou, ve dvou metrech už chyba dosahuje řádově centimetrů a senzor má tendenci zobrazovat nižší než skutečnou vzdálenost. Neprokázala se závislost na okolním osvětlení. Ačkoliv pro různé podmínky vyšly odlišné střední hodnoty vzdálenosti, nelze mezi nimi vzhledem k rozsahu testů najít trend.

b) Šum hloubkového senzoru

Otestování šumu se skládalo ze dvou dílčích testů. První z nich ověřil závislost šumu na materiálu snímaného povrchu. Za denního světla se provedlo měření na deskách ze dřeva, papíru a plastu, vzdálených 800 mm od kamery. Z 500 vzorků se spočítala směrodatná odchylka. Rozdíl mezi papírem a plastem je malý, na dřevě je šum menší. Příčinou může být kromě charakteru povrchu i jeho barva, avšak odrazivost pro různé barvy testována nebyla.

<i><b>Materiál</b></i>	<i><b>Skutečná vzdálenost [mm]</b></i>	<i><b>Směrodatná odchylka [mm]</b></i>
<i>Plast</i>	800	3,6
<i>Papír</i>	800	3,8
<i>Dřevo</i>	800	2,0

Tab. 16: Směrodatná odchylka různých materiálů



Obr. 53: Snímání plochy z plastu, papíru a dřeva

<i><b>Podmínky</b></i>	<i><b>Skutečná vzdálenost [mm]</b></i>	<i><b>Směrodatná odchylka [mm]</b></i>
<i>Denní světlo</i>	800	5,1
<i>Umělé osvětlení</i>	800	4,1
<i>Šero</i>	800	2,5
<i>Denní světlo</i>	2000	38,3
<i>Umělé osvětlení</i>	2000	24,9
<i>Šero</i>	2000	32,1

Tab. 17: Směrodatná odchylka za různých podmínek

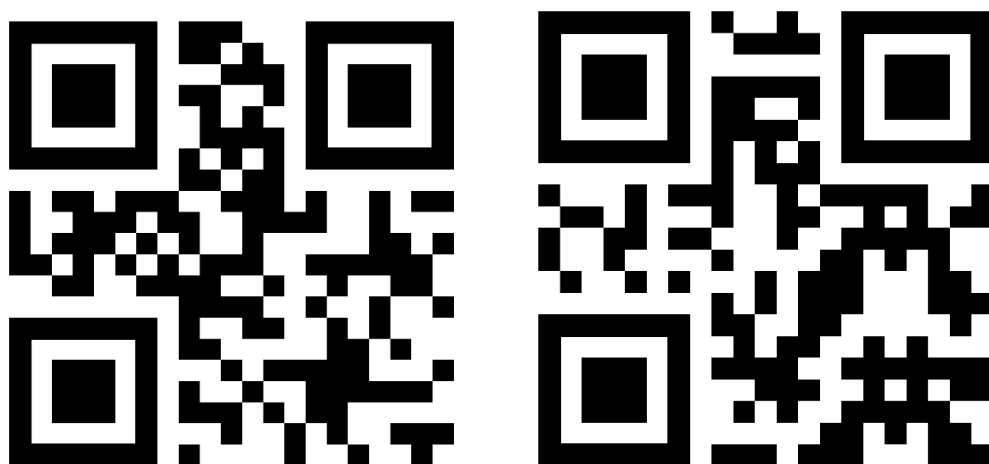
Pro druhý test byla využita stejná data jako v bodě a). Směrodatné odchylky neukazují jednoznačný vliv osvětlení na šum měření hloubky. Potvrdila se hypotéza, že s rostoucí vzdáleností roste i velikost šumu. Závěrem tohoto testu je, že vzhledem k vzdálenosti objektů lze uvažovat s přesností v rozmezí deseti milimetrů až tří centimetrů.

## 6.2. Volba značek

Dostáváme se k bodu, kde se začíná rozhodovat o tom, jakým způsobem bude snímán pracovní prostor. Myšlenka je taková, že se kamera umístí nad něj a bude tak vidět všechno potřebné shora. V prostoru by se měla nacházet značka, která určí, zda je kamera nad levou, nebo pravou rukou, navíc by mohla sloužit k určení počátku souřadného systému. Dále by měly být na rukách umístěny značky, které budou sloužit k určení polohy robota.

### 6.2.1. QR kód

QR kódy jsou dvoudimenzionální obrazce, které obsahují určitý text, často například odkaz na webovou stránku. Jsou populární kvůli jednoduchosti práce s nimi, k jejich vytvoření lze využít online generátory a snímat je lze různými detektory, například OpenCV má v sobě funkci sloužící přímo k detekci informací o kódu. Ta získá kromě obsahu QR kódu i informace o souřadnicích rohových bodů a binární obraz detekovaného kódu.



*Obr. 54: QR kódy - 1 (vlevo), 2 (vpravo)*

Na později specifikovaných místech jsou umístěny výše zobrazené kódy. Kód vlevo nese jako informaci číslici 1. Pokud ji script detekuje, ví, že je kamera nad levou rukou. Pokud detekuje 2, ví, že se nachází nad pravou rukou a podle toho se inicializují odpovídající funkcionality. Podle souřadnic kódu se určí i počátek souřadného systému.

### 6.2.2. Koule

Koule budou umístěny ve vhodných pozicích na rukách a budou sloužit k určování natočení jednotlivých vazeb. Využily se, protože je lze snadno identifikovat v libovolné viditelné poloze, narozdíl od 2D obrazců a kódů nehrozí, že by při určitých úhlech nemusely být detekovatelné.

#### a) Rozměry

Průměr koule je kompromisem mezi dvěma požadavky, a to dostatečnou velikostí, aby nebyl pro kameru problém kouli identifikovat, a naopak ne přílišnou velikostí, aby na konstrukci nezavazela a neomezovala robota při pohybu. Finální průměr činí 60 mm.

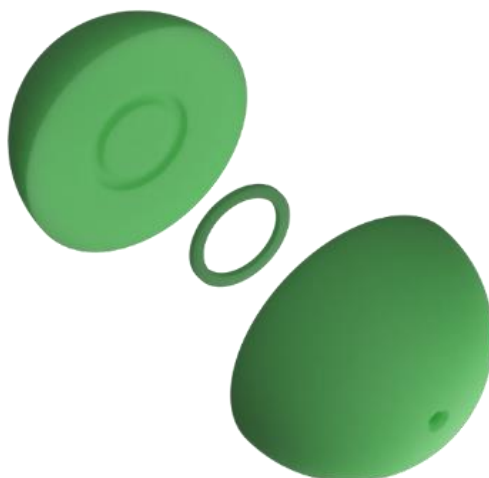
#### b) Barva

Barva by měla být taková, aby byla co nejmenší šance záměny s jiným objektem. Což vylučuje použití například černé a hnědé. Nabízí se výraznější barvy, například červená, fialová, zelená. Neexistuje optimální barva, svému účelu poslouží větší množství kombinací. Volba se odvíjela od dostupného filamentu, proto jsou koule světle zelené.

#### c) Materiál

Jako prototyp byly využity tenisové míčky, pro finální verzi však bylo potřeba značky pevněji a přesněji ukotvit, proto se vymodelovaly a následně vytiskly. Aby šlo díl

vytisknout, skládá se ze dvou polokoulí a obruče pro snadnější a přesnější slepení. Na jedné z polokoulí je díra, do které se vyřezal závit k ukotvení míčku do dílu z pásoviny.



*Obr. 55: Součásti koule*

### **6.3. Umístění značek na konstrukci**

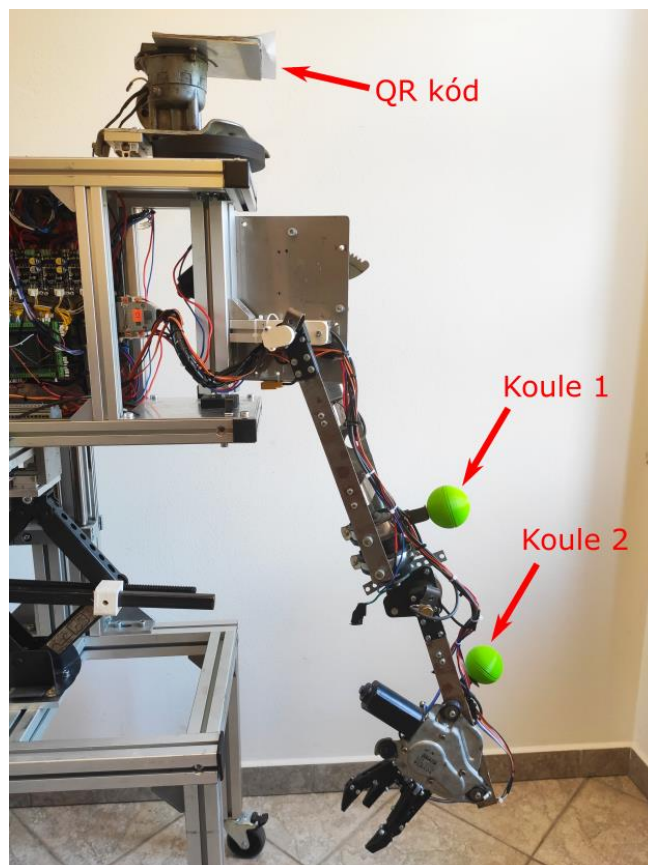
QR kódy by se měly nacházet poblíž počátku souřadného systému. Co se týče koulí, je třeba jich na každou ruku umístit tolik, aby šlo natočení všech vazeb určit jednoznačně algebraicky. Je však nutno brát ohled na různé singulární polohy a další faktory, které by mohly znemožňovat stanovení úhlů, případně razantně snižovat přesnost estimace.

#### **6.3.1. Levá ruka**

Střed QR kódu se nachází nad osu rotace první vazby, protože zde mají počátek  $x$  a  $y$  souřadnice uvažovaného souřadného systému. Levá ruka má čtyři stupně volnosti, k jednoznačnému určení její polohy postačují dva míčky. Pomocí prvního lze určit natočení  $\varphi_1$  a  $\varphi_2$ , pomocí druhého  $\varphi_3$  a  $\varphi_4$ .



*Obr. 56: Pracovní prostor – pohled shora*



*Obr. 57: Pracovní prostor – pohled zepředu*

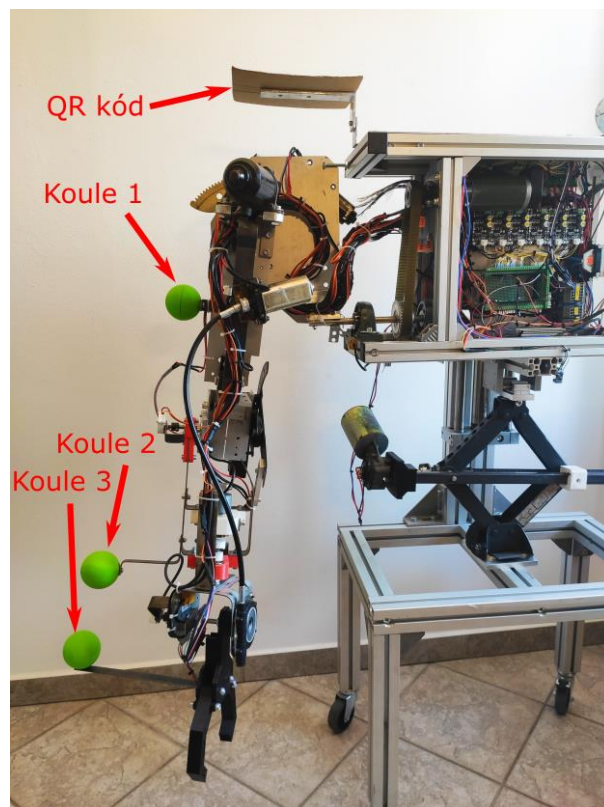
### **6.3.2. Pravá ruka**

QR kód je umístěn na plošině tak, aby kvůli němu nebylo omezeno zorné pole kamery a nebyl překážkou pro detekci jiných objektů a také překážkou v pohybu. Na ruku se umístily tři míčky, každý bude sloužit k určení dvou kloubových souřadnic. Jsou umístěny tak, aby byla zajištěna jejich přímá viditelnost pro obvyklé pohyby.



*Obr. 58: Pracovní prostor – pohled shora*





Obr. 59: Pracovní prostor – pohled zepředu

## 6.4. Identifikace značek

### 6.4.1. Čtečka QR

Pro identifikaci QR kódu slouží OpenCV třída `QRCodeDetector()`, která obsahuje funkci `detectAndDecode(img)`, jejímž vstupem je obraz, ve kterém má být kód detekován. Funkce má tři výstupy. První, `data`, obsahuje text kódu. Druhá, `bbox`, obsahuje čtyři rohové body nalezený kódu. Třetí, `rectifiedImage`, obsahuje QR kód jako binární obrazec.

### 6.4.2. Houghova transformace

Princip této metody je popsán v kapitole 3.5.3, zde si popíšeme samotnou implementaci kódu. Nejprve se obrázek převede do černobílé škály funkcí `cv2.cvtColor(...)`, poté se doporučuje mírné rozmazání obrazu kvůli snížení počtu falešných pozitiv, například funkcí `medianBlur(img, 7)`. Takto upravený snímek již může být využit k samotné detekci funkcí:

```
cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, minDist, par1, par2, minR, maxR)
```

, kde `img`... černobílý snímek

`cv2.HOUGH_GRADIENT`... metoda detekce, aktuálně jediná možná

`minDist`... minimální vzdálenost středů objektů

`par1, par2`... parametry detekce

`minR, maxR`... minimální a maximální poloměry detekovaných objektů

Klíčové je správné naladění parametrů `par1`, `par2`. Neexistuje univerzální metoda, jak je spočítat, naladily se tedy ručně.

Parametr	Hodnota [-]
<code>par1</code>	220
<code>par2</code>	40

*Tab. 18: Parametry `cv2.HoughCircles`*



*Obr. 60: Detekce míčků Houghovou transformací*

Detekce kruhových objektů využitím Houghovy transformace vykazuje vysokou úspěšnost v detekci míčků umístěných na robotovi, a to s nízkou závislostí na osvětlení. Velkou nevýhodou je snadná zaměnitelnost za libovolný jiný kulatý objekt v zorném poli.

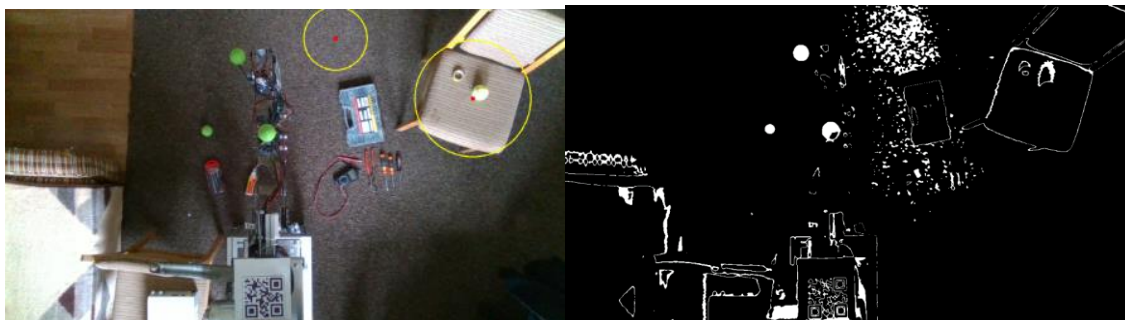
### 6.4.3. RGB

Detekce pomocí barevného profilu spočívá v určení spodní a horní meze každé ze složek a analýze pixelů, zda všechny jeho složky leží v požadovaném intervalu. OpenCV má pro tento účel implementovanou funkci `cv2.inRange(img, lower, upper)`, kde `lower` je spodní hranice daná třemi hodnotami (R, G, B) a `upper` horní hranice. Po jejím aplikování se použije funkce `cv2.findContours(...)`, která najde „ostrovy“ pozitivních nálezů.

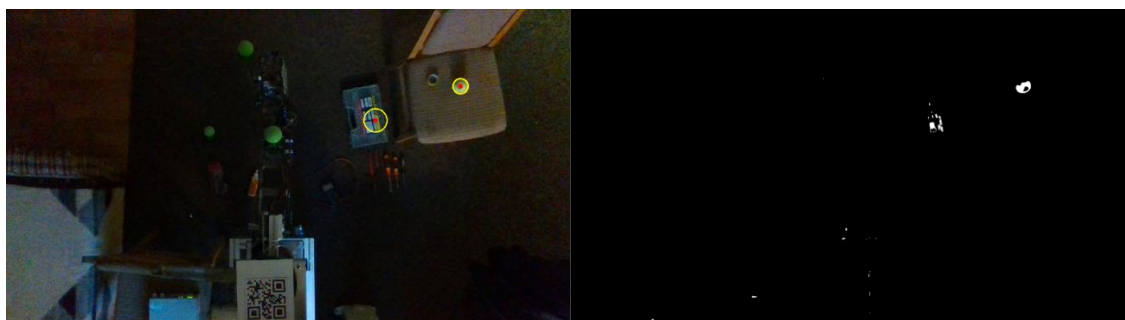
Parametry se ladily ručně, zobrazovaly se pozitivní nálezy a hledala se optimální kombinace všech šesti parametrů. Řešení se otestovalo za třech světelných podmínek, a to za denního světla, večerního šera a umělého osvětlení.

Parametr	R – červená složka	G – zelená složka	B – modrá složka
<b>Dolní mez</b>	35	110	55
<b>Horní mez</b>	145	255	150

*Tab. 19: Mezní parametry RGB detekce objektů*



*Obr. 61: RGB detekce za denního světla, pozitivní nálezy*



*Obr. 62: RGB detekce za šera, pozitivní nálezy*



*Obr. 63: RGB detekce při umělém osvětlení, pozitivní nálezy*

Z výše umístěných obrázků plyne nedostatečnost metody. Obstojné naladění koeficientů pro dané světelné podmínky se jeví při změně osvětlení jako nepoužitelné. Plyne to z podstaty RGB modelu, kde všechny tři složky udávají intenzitu odpovídající barvy. Při zvyšující se intenzitě osvětlení objektu roste hodnota všech složek spektra. Pro praktické využití tento způsob nesplňuje požadavky na přesnost a spolehlivost.

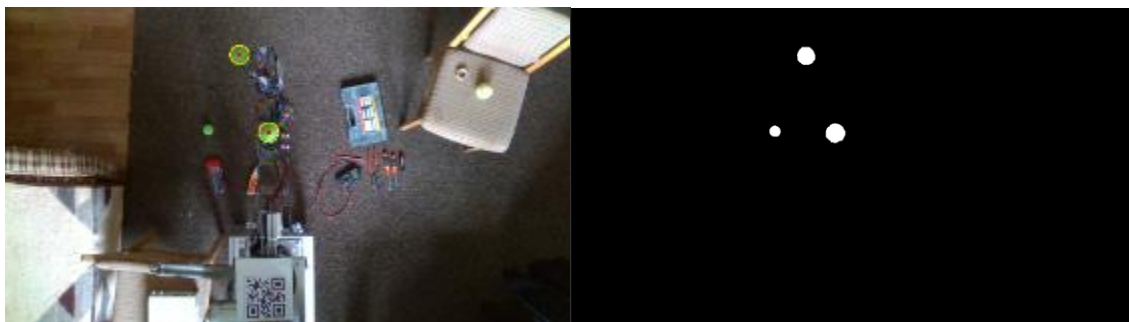
#### 6.4.4. HSV

Vzhledem k nedostatečnosti předchozího modelu se nabízí využití velmi podobné detekce, avšak v HSV barevném spektru. To pracuje s jasnem a odstínem barvy a může tedy redukovat závislost na světelných podmínkách. Kód je stejný jako v předchozím případě, jedinou změnu tvoří převedení snímku do HSV příkazem `cv2.cvtColor(img, cv2.COLOR_BGR2HSV)`.

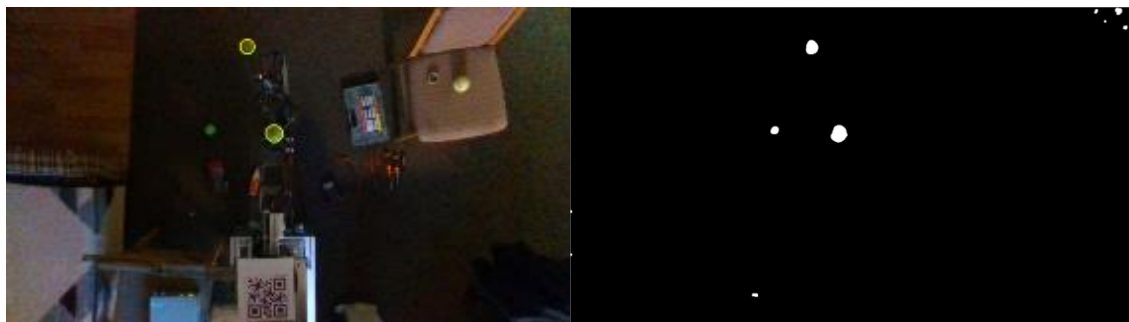
Parametr	H – odstín	S – sytost	V – jas
Dolní mez	50	80	130
Horní mez	90	255	255

*Tab. 20: Mezní parametry HSV detekce objektů*

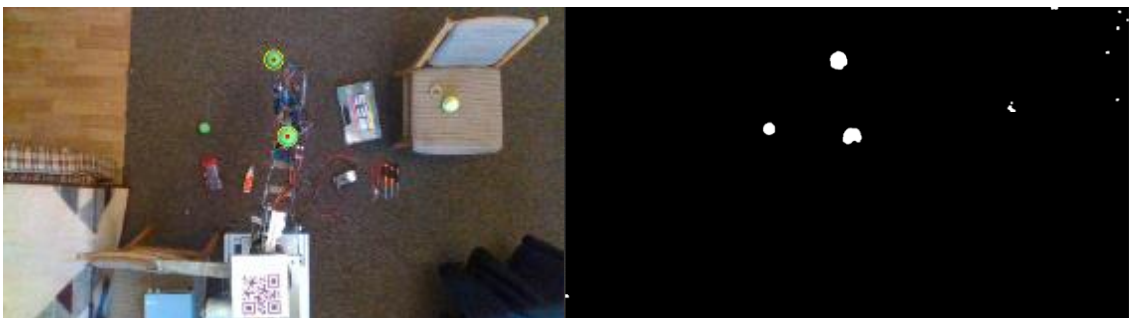




*Obr. 64: HSV detekce za denního světla, pozitivní nálezy*



*Obr. 65: HSV detekce za šera, pozitivní nálezy*



*Obr. 66: HSV detekce při umělém osvětlení, pozitivní nálezy*

Na první pohled je viditelné, jak velké zlepšení tato metoda oproti RGB přináší. Je schopná spolehlivě detekovat všechny tři míčky za většiny světelných podmínek, zatímco šum je omezen téměř na minimum. Z tohoto důvodu se tato metoda prohlásila za nejlepší a v následujících kapitolách bude pracováno pouze s ní.

## 7. Vizualní zpětná vazba

Jelikož se tvorbou dopředné i inverzní kinematiky podrobněji zabývá práce [1], využívají se zde její části podle aktuální potřeby. Všechny značky jsou rozmístěny podle kapitoly 6.3 a jsou detekovány pomocí HSV metody, popsané v kapitole 6.4.4. Tato kapitola obsahuje odvozené rovnice, které se používají pro určování úhlů na obou rukou. Následně bylo ověřeno, zda jsou správné, a to testem, kde se porovnaly spočtené hodnoty s reálnými hodnotami z potenciometrů.

Na úvod si ukažme obecné tvary transformačních matic, protože se v této kapitole počítá dopředná kinematika jejich využitím. Pro osy x, y a z mají následující tvar:

$$T_x(\varphi, \begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & \cos(\varphi) & -\sin(\varphi) & y \\ 0 & \sin(\varphi) & \cos(\varphi) & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

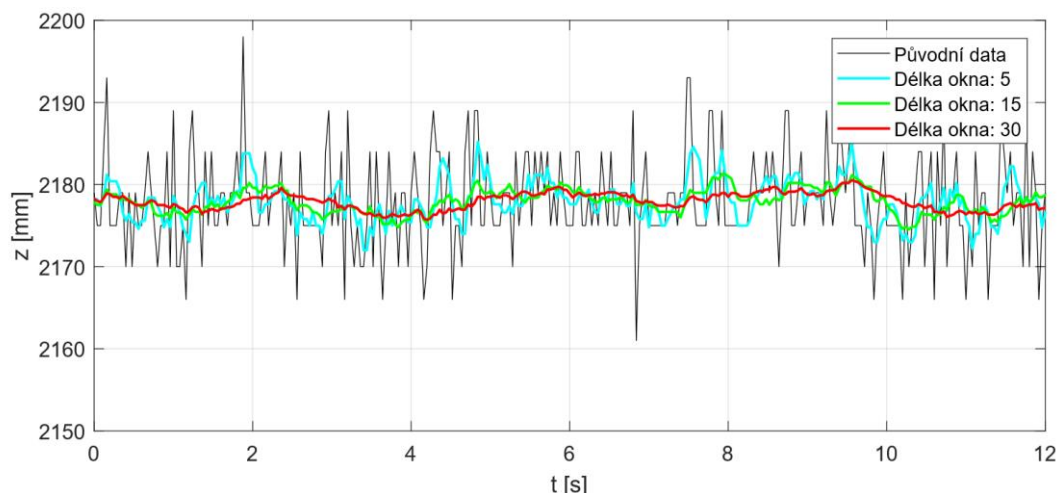
$$T_y(\varphi, \begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{pmatrix} \cos(\varphi) & 0 & \sin(\varphi) & x \\ 0 & 1 & 0 & y \\ -\sin(\varphi) & 0 & \cos(\varphi) & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

$$T_z(\varphi, \begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & 0 & x \\ \sin(\varphi) & \cos(\varphi) & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

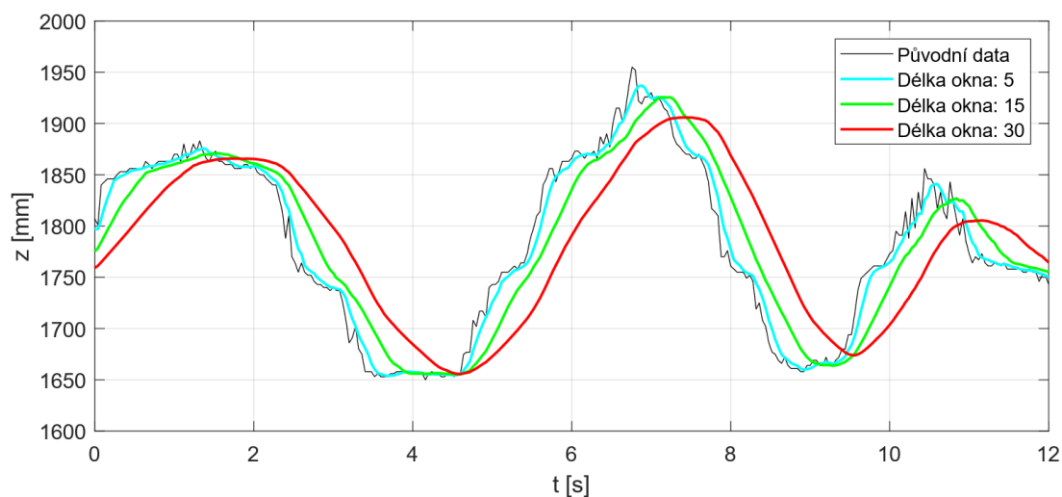
Matice 3x3 vlevo nahoře značí rotaci o úhel  $\phi$  okolo odpovídající osy, pokud by se využívala pouze tato 3x3 matice, bude označena jako  $R_x, R_y, R_z$ . Vektor  $[x \ y \ z]^T$  je translace bodu rotace oproti předchozímu bodu.

### 7.1. Filtrace dat

V kapitole 6.1.3 byl otestován senzor Realsense při reálném použití. Závěrem bylo, že šum hloubkového senzoru může mít významný vliv na kvalitu estimace úhlů, proto byl do kódu implementován *moving average filter*, který počítá aritmetický průměr posledních  $x$  vzorků, přičemž  $x$  je délka filtračního okna. Filtrují se kartézské souřadnice detekovaných objektů a délka filtračního okna se mění v závislosti na konkrétním použití. Experiment na obr. 67 a 68 zobrazuje testování různých oken při vzorkovací frekvenci 25 Hz. Z něj plyne, že se musí najít vhodný kompromis mezi šumem střední hodnoty a zpožděním.



Obr. 67: Moving average filtr – rovina



Obr. 68: Moving average filtr – změna výšky

## 7.2. Levá ruka

### 7.2.1. Souřadný systém

Pravotočivý kartézský souřadný systém (viz kinematické schéma obr. 69 zeleně) má z osu má totožnou s osou první rotace, kladná x osa směřuje dopředu a kladná y osa doprava při pohledu zepředu. Počátek z osy je ve výšce, ve které dochází k rotaci kolem kloubové souřadnice  $\varphi_2$ . Následující rovnice udávají přepočítání ze souřadného systému kamery do souřadného systému manipulátoru.

$$x = x' - x'_0 \quad (8)$$

$$y = -y' + y'_0 \quad (9)$$

$$z = -z' + z'_0 + z_0 \quad (10)$$

, kde  $x, y, z \dots$  kartézské souřadnice v souřadnicovém systému manipulátoru

$x', y', z' \dots$  kartézské souřadnice v souřadnicovém systému kamery

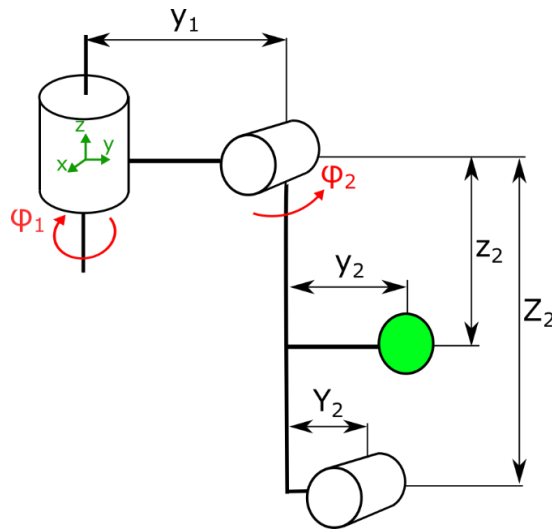
$x'_0, y'_0, z'_0 \dots$  kartézské souřadnice QR kódu v souřadnicovém systému kamery

$z_0 = 335 \text{ mm}$  ... hloubka mezi QR kódem a středem souřadnicového systému manipulátoru

### 7.2.2. Vzorce

Předpokládáme, že máme detekovány oba míčky a jejich polohu přepočtenou do vhodného souřadného systému. Označme  $(m_{1x}, m_{1y}, m_{1z})$  kartézské souřadnice středu prvního míčku a obdobně  $(m_{2x}, m_{2y}, m_{2z})$  střed druhého míčku. V tuto chvíli je zapotřebí určit, který detekovaný objekt odpovídá prvnímu a který druhému míčku s ohledem na jejich umístění na konstrukci. Toho lze docílit porovnáním x a y souřadnic obou. Musí platit (11), pokud neplatí, pořadí míčků se prohodí.

$$\frac{m_{1x}}{m_{1z}} > \frac{m_{2x}}{m_{2y}} \quad (11)$$



Obr. 69: Kinematické schéma detekce prvního míčku

Parametr	Rozměr [mm]
$y_1$	124
$y_2$	105
$z_2$	290
$Y_2$	35
$Z_2$	390

Tab. 21: Rozměry konstrukce

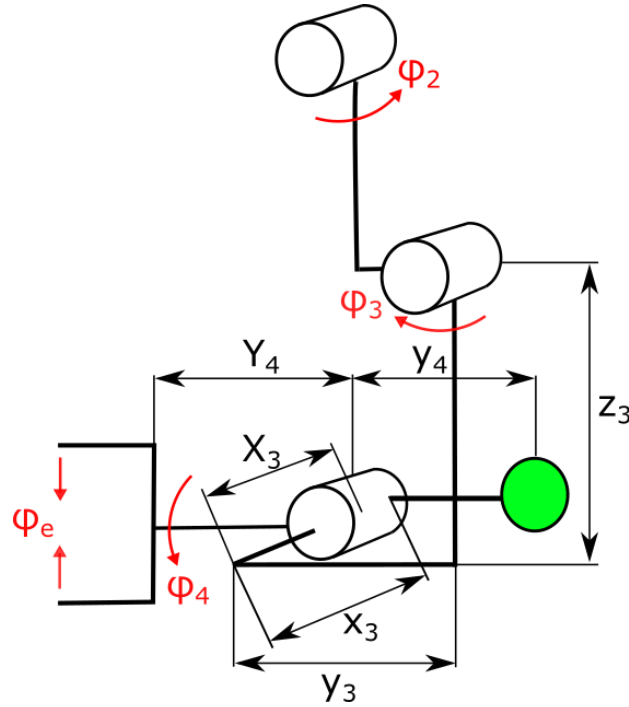
První míček umožňuje určit úhly vazeb  $\phi_1$  a  $\phi_2$ . Natočení kloubové souřadnice  $\phi_1$  závisí na poměru x a y souřadnic míčku, natočení  $\phi_2$  se určí z hloubky míčku vůči předchozímu bodu a je dáno rovnicí (13).

$$\phi_1 = \tan^{-1}\left(\frac{m_{1x}}{m_{1y}}\right) \quad (12)$$

$$\phi_2 = \cos^{-1}\left(\frac{-m_{1z}}{\sqrt{y_2^2 + z_2^2}}\right) - \tan^{-1}\left(\frac{y_2}{z_2}\right) \quad (13)$$

Známe-li první dvě natočení, můžeme pomocí rotačních matic získat souřadnice bodu, ve kterém probíhá rotace třetího aktuátoru, a to podle rovnice (14). Tento bod označme jako  $P_1$  a jeho složky v osách souřadného systému jako  $(p_{1x}, p_{1y}, p_{1z})$ .

$$P_1 = \begin{pmatrix} p_{1x} \\ p_{1y} \\ p_{1z} \\ 1 \end{pmatrix} = T_z \left( -\varphi_1, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot T_x \left( -\varphi_2, \begin{bmatrix} 0 \\ y_1 \\ 0 \end{bmatrix} \right) \cdot \begin{pmatrix} 0 \\ Y_2 \\ -Z_2 \\ 1 \end{pmatrix} \quad (14)$$



Obr. 70: Kinematické schéma detekce druhého míčku

Parametr	Rozměr [mm]
$x_3$	100
$y_3$	50
$z_3$	280
$y_4$	141
$X_4$	35
$Y_4$	150

Tab. 22: Rozměry konstrukce

Geometrie pro určení natočení  $\varphi_3$  a  $\varphi_4$  vychází z kinematického schématu na obr. 70. Pro zjednodušení následujících výpočtů provedeme korekci rozměru  $y_4$  vůči uvažovanému středu osy rotace vazeb  $\varphi_3$  a  $\varphi_4$  podle rovnic (15) a (16). Tímto se tohoto rozměru zbavíme a v dalších rovnicích s ním již nemusíme počítat.

$$m'_{1x} = m_{1x} + y_4 \cdot \cos(\varphi_1) \quad (15)$$

$$m'_{1y} = m_{1y} - y_4 \cdot \sin(\varphi_1) \quad (16)$$

Rozdíly v souřadnicích druhého míčku oproti bodu  $P_I$  označíme jako (17)-(19). Celková vzdálenost mezi těmito dvěma body  $\Delta X$  je rovna normě těchto složek.

$$\Delta x = m'_{2x} - p_{1x} \quad (17)$$

$$\Delta y = m'_{2y} - p_{1y} \quad (18)$$

$$\Delta z = m_{2z} - p_{1z} \quad (19)$$

$$\Delta X = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (20)$$

Již odvozené rovnice pro výpočet zbylých dvou úhlů vypadají následovně:

$$\varphi_3 = \tan^{-1}\left(\frac{\Delta z}{\sqrt{\Delta x^2 + \Delta y^2}}\right) - \tan^{-1}\left(\frac{y_4}{z_3}\right) + \cos^{-1}\left(\frac{z_3^2 + y_4^2 + \Delta X^2 - y_4^2}{2 \cdot \sqrt{z_3^2 + y_4^2} \cdot \Delta X}\right) + \frac{\pi}{2} + \varphi_2 \quad (21)$$

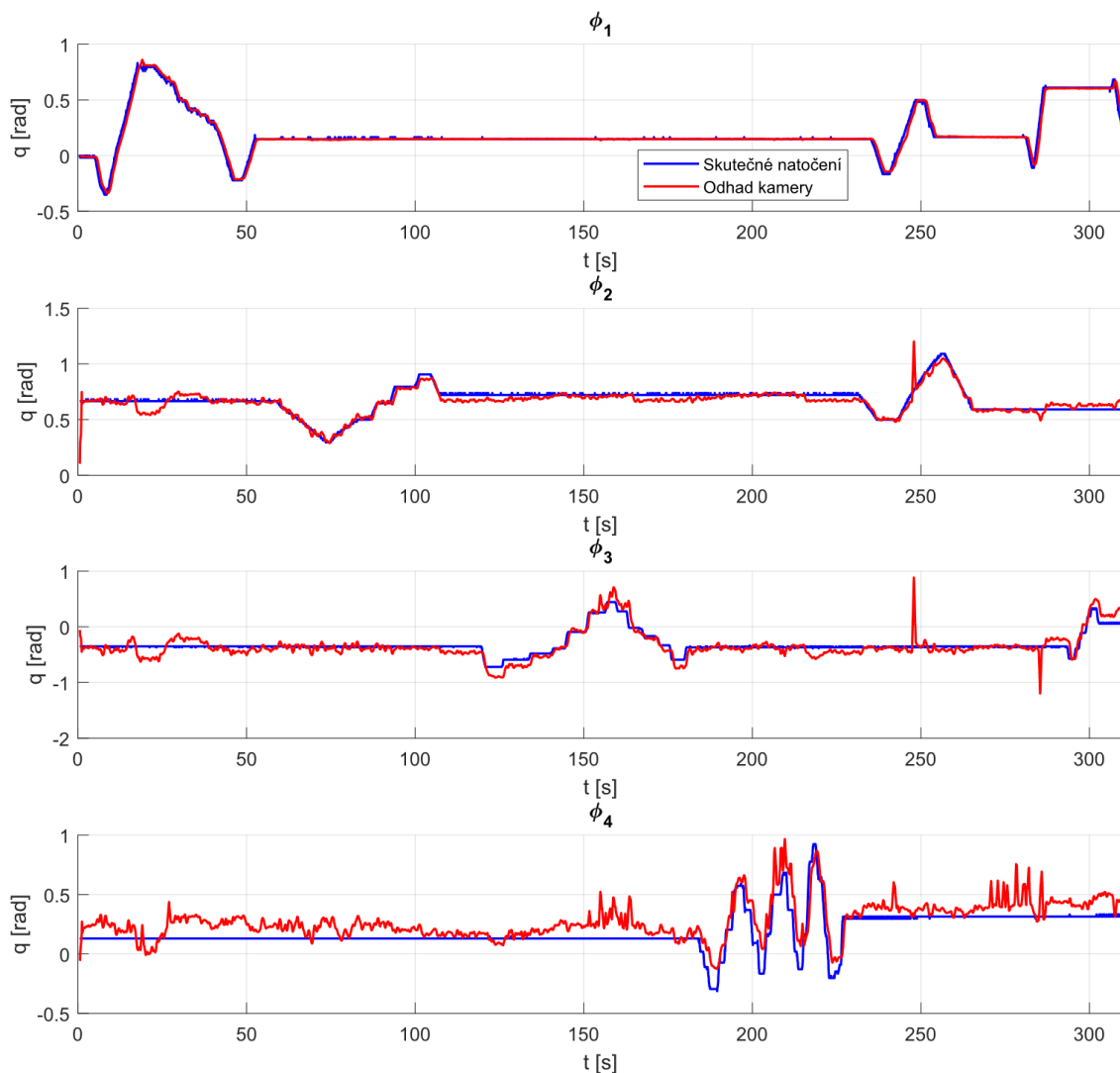
$$\varphi_4 = -\cos^{-1}\left(\frac{z_3^2 + 2 \cdot y_4^2 - \Delta X^2}{2 \cdot \sqrt{z_3^2 + y_4^2} \cdot y_4}\right) + \tan^{-1}\left(\frac{z_3}{y_4}\right) \quad (22)$$

Známe-li všechny kloubové souřadnice, lze využitím dopředné kinematiky a rotace kolem odpovídajících os o odpovídající úhel získat kartézské souřadnice koncového bodu efektoru  $P$ .

$$P = T_z\left(-\varphi_1, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right) \cdot T_x\left(-\varphi_2, \begin{bmatrix} 0 \\ y_1 \\ 0 \end{bmatrix}\right) \cdot T_x\left(\varphi_3, \begin{bmatrix} 0 \\ Y_2 \\ -Z_2 \end{bmatrix}\right) \cdot T_x\left(-\varphi_4, \begin{bmatrix} -X_3 \\ -y_3 \\ -z_3 \end{bmatrix}\right) \cdot \begin{pmatrix} 0 \\ -Y_4 \\ 0 \\ 1 \end{pmatrix} \quad (23)$$

### 7.2.3. Výsledky

Správnost rovnic byla ověřena postupným pohybem všemi pohony ve většině rozsahu tak, aby se získal komplexní přehled o kvalitě metody. Data se filtrovala *moving average* filtrem s délkou okna  $x = 10$ . Grafy na obr. 71 dokazují, že použité rovnice správně počítají jednotlivé úhly, avšak je patrné, že se s rostoucím číslem vazby se snižuje přesnost. To je dáno nepřesnostmi ve vytvořené konstrukci a také více se projevujícím šumem.



Obr. 71: Porovnání skutečného a odhadovaného natočení, levá ruka

### 7.3. Pravá ruka

#### 7.3.1. Souřadný systém

Souřadný systém této ruky (zeleně na obr. 72) je definován z osou, která směřuje vertikálně a její kladná výchylka je směrem dolů. Y osa je totožná s osou rotace první vazby, x osa směřuje dopředu a je umístěna na okraji těla robota. Systém se stejně jako v případě levé ruky musí transformovat do systému manipulátoru následujícími rovnicemi:

$$x = -x' + x'_0 \quad (24)$$

$$y = -y' + y'_0 + 0,12 \quad (25)$$

$$z = z' - z'_0 - z_0 \quad (26)$$

, kde  $x, y, z, \dots$  kartézské souřadnice v souřadnicovém systému manipulátoru

$x', y', z', \dots$  kartézské souřadnice v souřadnicovém systému kamery

$x'_0, y'_0, z'_0, \dots$  kartézské souřadnice QR kódu v souřadnicovém systému kamery

$z_0 = 355 \text{ mm}$  ... hloubka mezi QR kódem a středem souřadnicového systému manipulátoru

### 7.3.2. Vzorce

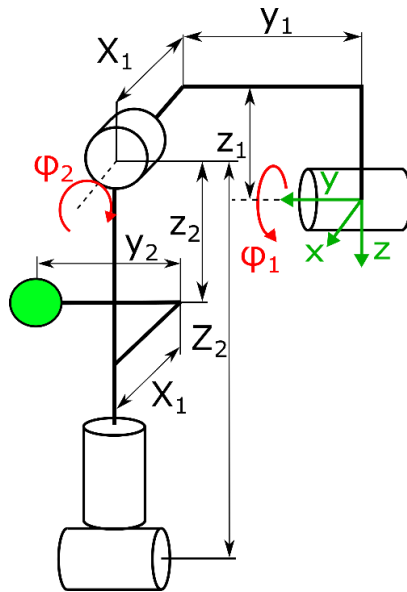
Prvním krokem je přiřazení detekovaných objektů ke konkrétním míčkům. Označme si míčky jako  $M_1 = (m_{1x}, m_{1y}, m_{1z})$ ,  $M_2 = (m_{2x}, m_{2y}, m_{2z})$ ,  $M_3 = (m_{3x}, m_{3y}, m_{3z})$ , index 1 značí míček nejbližší k tělu. Pro první míček platí, že je nejbližší bodu, který se nachází pod osou druhé rotace a na ose první rotace, rovnicemi zapsáno (27) a (28). Pokud to splněno není, prohodí se míčky mezi sebou.

$$\sqrt{m_{1x}^2 + m_{1y}^2 + (m_{1z} - y_1)^2} < \sqrt{m_{2x}^2 + m_{2y}^2 + (m_{2z} - y_1)^2} \quad (27)$$

$$\sqrt{m_{1x}^2 + m_{1y}^2 + (m_{1z} - y_1)^2} < \sqrt{m_{3x}^2 + m_{3y}^2 + (m_{3z} - y_1)^2} \quad (28)$$

Zbývá zjistit, zda nejsou zbylé dva míčky naopak. Vzhledem k rozsahu pohybu jednotlivých vazeb by měl být druhý míček vždy blíže k prvnímu než třetí.

$$\sqrt{(m_{2x} - m_{1x})^2 + (m_{2y} - m_{1y})^2 + (m_{2z} - m_{1z})^2} < \sqrt{(m_{3x} - m_{1x})^2 + (m_{3y} - m_{1y})^2 + (m_{3z} - m_{1z})^2} \quad (29)$$



Obr. 72: Kinematické schéma prvního míčku

Parametr	Rozměr [mm]
$y_1$	179
$z_1$	137
$X_1$	35
$y_2$	91
$z_2$	104
$Z_2$	310

Tab. 23: Rozměry konstrukce



Následuje stanovení natočení prvních dvou pohonů  $\varphi_1$  a  $\varphi_2$  podle rovnic (30)-(32).

$$\varphi_2 = \sin^{-1}\left(\frac{\Delta y}{\sqrt{y_2^2 + z_2^2}}\right) - \tan^{-1}\left(\frac{y_2}{z_2}\right), \text{ když } \sqrt{m_{1x}^2 + m_{1z}^2} \leq z_1 \quad (30)$$

$$\varphi_2 = \pi - \sin^{-1}\left(\frac{\Delta y}{\sqrt{z_2^2 + y_2^2}}\right) - \tan^{-1}\left(\frac{y_2}{z_2}\right), \text{ jindy} \quad (31)$$

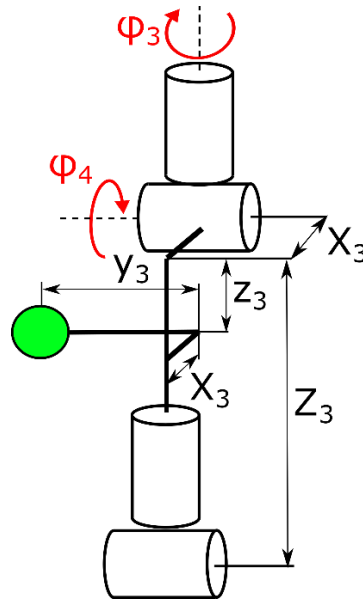
$$\varphi_1 = -\sin^{-1}\left(\frac{m_{1x}}{z_1 - z_2 \cdot \cos(\varphi_2) + y_2 \cdot \sin(\varphi_2)}\right) \quad (32)$$

Po určení prvních dvou natočení lze dopřednou kinematikou dopočítat bod, ve kterém jsou realizovány třetí a čtvrtá rotace. Zároveň si jako  $\Delta X'_2$  označíme vzdálenost druhého míčku od tohoto bodu. Pro zjednodušení dalších operací se tento rozdíl orotoval o záporné hodnoty úhlů kolem odpovídajících os. Tím se dostane do stavu, který odpovídá rozdílu mezi těmito body v případě, že by první dva úhly byly nulové. To zjednoduší další rovnice, protože se v nich již  $\varphi_1$  a  $\varphi_2$  nebudou vyskytovat.

$$P_1 = T_y\left(\varphi_1, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right) \cdot T_x\left(\varphi_2, \begin{bmatrix} X_1 \\ y_1 \\ -z_1 \end{bmatrix}\right) \cdot \begin{pmatrix} 0 \\ 0 \\ z_2 \\ 1 \end{pmatrix} \quad (33)$$

$$\Delta P'_2 = M_2 - P_1 \quad (34)$$

$$\Delta P_2 = R_x(-\varphi_2) \cdot R_y(-\varphi_1) \cdot \Delta P'_2 \quad (35)$$



Obr. 73: Kinematické schéma druhého míčku

<sup>1</sup> Z  $P_1$  se berou pouze první 3 složky  $p_{1x}, p_{1y}, p_{1z}$

Parametr	Rozměr [mm]
$y_3$	175
$z_3$	230
$X_3$	65
$Z_3$	300

Tab. 24: Rozměry konstrukce

Natočení  $\varphi_3$  a  $\varphi_4$  vychází z rovnic (36) a (37).

$$\varphi_3 = \sin^{-1} \left( \frac{y_3}{\sqrt{\Delta x_2^2 + \Delta y_2^2}} \right) + \tan^{-1} \left( \frac{\Delta x_2}{\Delta y_2} \right) - \frac{\pi}{2} \quad (36)$$

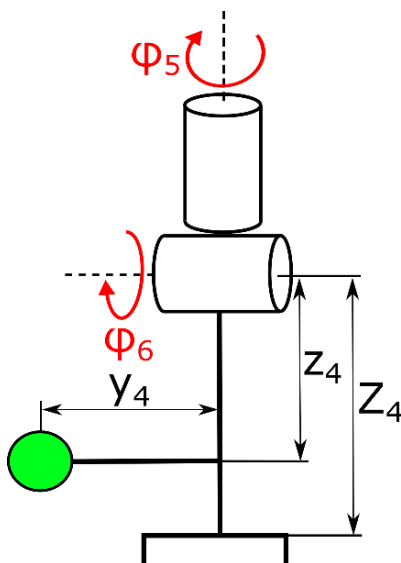
$$\varphi_4 = \sin^{-1} \left( \frac{\sqrt{\Delta x_2^2 + \Delta y_2^2} - y_3^2}{z_3} \right) \quad (37)$$

Dopřednou kinematiku se spočítá bod, který představuje průsečík zbývajících dvou rotací. Stejně jako v předchozím případě stanovíme rozdíl mezi souřadnicemi posledního míčku a průsečíku os rotací. Tento rozdíl se orotuje o záporné úhly  $\phi_1$  až  $\phi_4$ .

$$P_2 = T_y \left( \varphi_1, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot T_x \left( \varphi_2, \begin{bmatrix} X_1 \\ y_1 \\ -z_1 \end{bmatrix} \right) \cdot T_z \left( \varphi_3, \begin{bmatrix} 0 \\ 0 \\ z_2 \end{bmatrix} \right) \cdot T_y \left( -\varphi_4, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot \begin{pmatrix} X_3 \\ 0 \\ z_3 \\ 1 \end{pmatrix} \quad (38)$$

$$\Delta P'_3 = M_3 - P_2 \quad (39)$$

$$\Delta P_3 = R_y(\varphi_4) \cdot R_z(-\varphi_3) \cdot R_x(-\varphi_2) \cdot R_y(-\varphi_1) \cdot \Delta P'_3 \quad (40)$$



Obr. 74: Kinematické schéma třetího míčku

<sup>1</sup> Z  $P_2$  se berou pouze první 3 složky  $p_{2x}, p_{2y}, p_{2z}$

Parametr	Rozměr [mm]
$y_4$	231
$z_4$	70
$Z_4$	220

Tab. 25: Rozměry konstrukce

Rovnice pro stanovení zbývajících dvou úhlů jsou prakticky totožné jako pro úhly  $\varphi_3$  a  $\varphi_4$ . Nevýhodou těchto dvou rovnic je nejednoznačnost řešení. V předchozím případě to problém nebyl, protože  $\varphi_4$  nemůže nabývat záporných hodnot, naopak  $\varphi_6$  se může pohybovat do kladného i záporného směru. Pak se do jednoho bodu dá kombinací  $\varphi_5$  a  $\varphi_6$  dostat dvěma způsoby. Omezme se však pouze na případy, kdy  $\varphi_6 > 0$ .

$$\varphi_5 = \sin^{-1} \left( \frac{y_4}{\sqrt{\Delta x_3^2 + \Delta y_3^2}} \right) + \tan^{-1} \left( \frac{\Delta x_3}{\Delta y_3} \right) - \frac{\pi}{2} \quad (41)$$

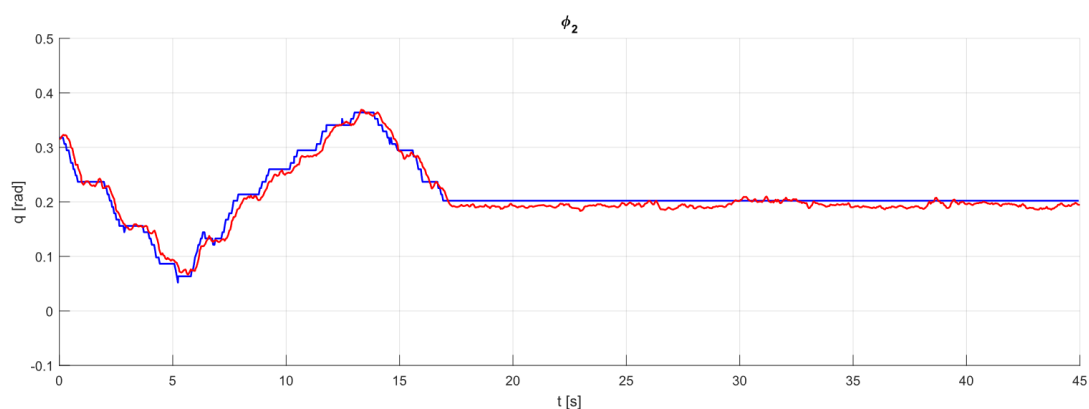
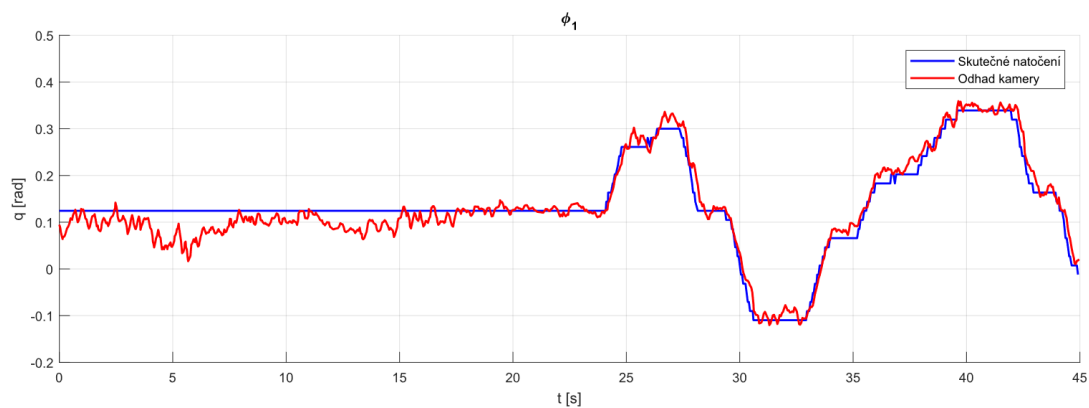
$$\varphi_6 = \sin^{-1} \left( \frac{\sqrt{\Delta x_3^2 + \Delta y_3^2} - y_4^2}{Z_4} \right) \quad (42)$$

Při znalosti všech rovnic je posledním krokem dopřednou kinematikou určit souřadnice koncového bodu efektoru v prostoru podle rovnice (43).

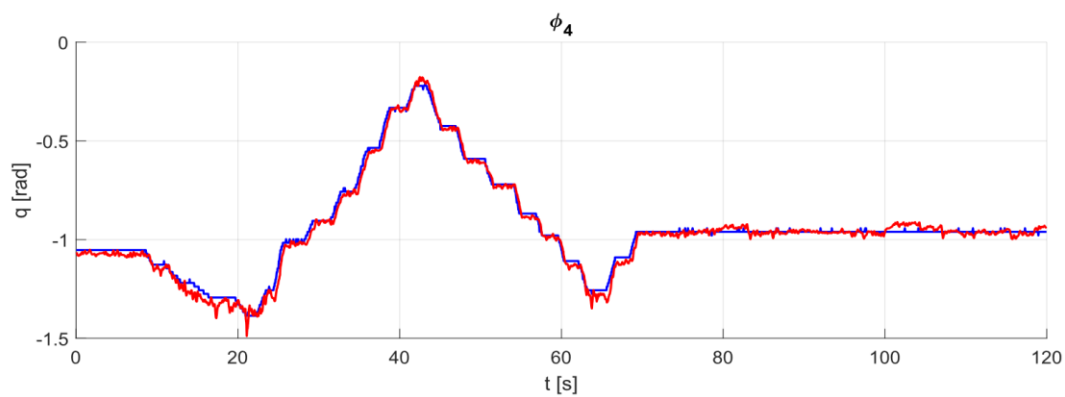
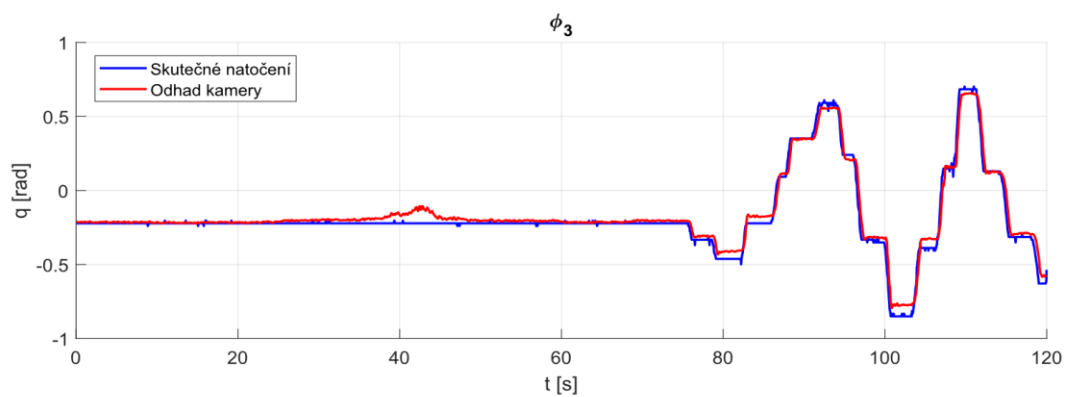
$$P = T_y \left( \varphi_1, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot T_x \left( \varphi_2, \begin{bmatrix} X_1 \\ y_1 \\ -z_1 \end{bmatrix} \right) \cdot T_z \left( \varphi_3, \begin{bmatrix} 0 \\ 0 \\ Z_2 \end{bmatrix} \right) \cdot T_y \left( -\varphi_4, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot T_z \left( \varphi_5, \begin{bmatrix} X_3 \\ 0 \\ Z_3 \end{bmatrix} \right) \cdot T_y \left( \varphi_6, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot \begin{pmatrix} 0 \\ 0 \\ Z_4 \\ 1 \end{pmatrix} \quad (43)$$

### 7.3.3. Výsledky

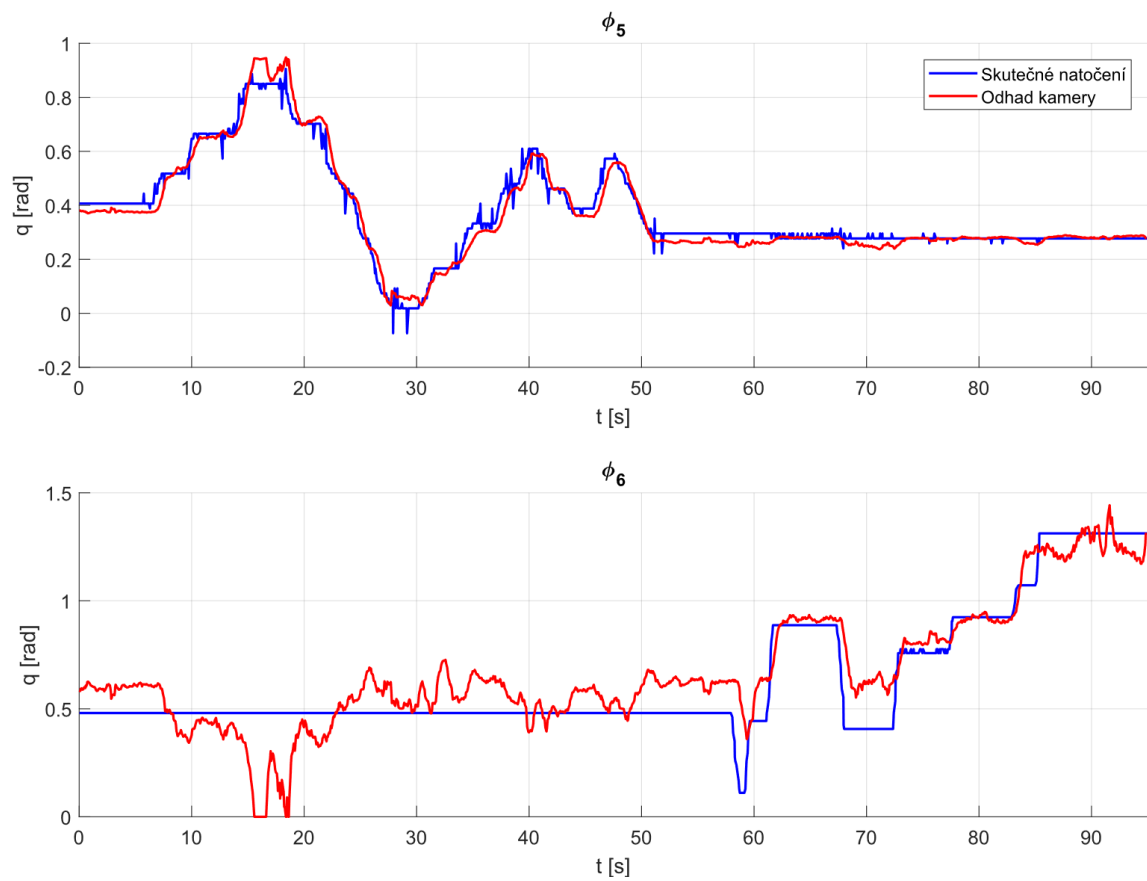
Ověřování správnosti rovnic probíhalo po jednotlivých míčcích, to znamená, že se určovaly vždy dvě kloubové souřadnice (obr. 75 až 77). Vzhledem k tomu, že průběhy odhadu i reálných dat mají stejný tvar, bylo prohlášeno, že rovnice jsou vytvořeny správně. Na druhou stranu se zejména u poslední kloubové souřadnice nepodařilo dosáhnout požadované přesnosti (obr. 77 dole), protože konstrukčně je toto rameno poměrně krátké a v kombinaci s přesností kamery a nepřesnostmi předchozích vazeb se objevuje značná odchylka.



Obr. 75: Estimace úhlů  $\phi_1$  a  $\phi_2$



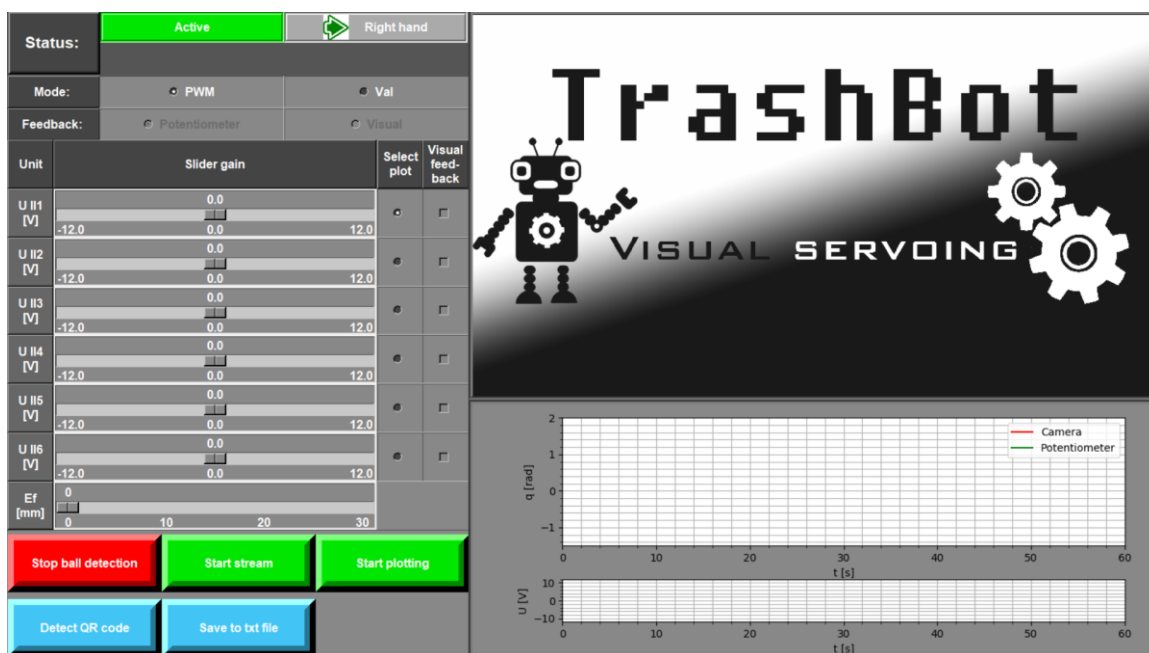
Obr. 76: Estimace úhlů  $\phi_3$  a  $\phi_4$



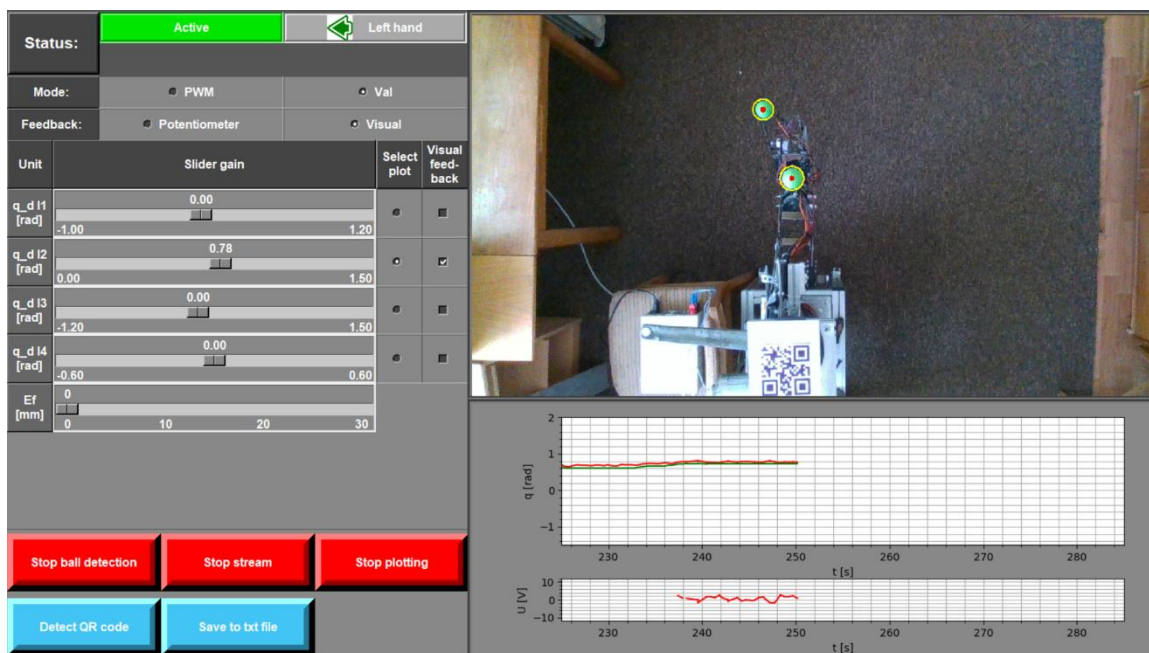
*Obr. 77: Estimace úhlů  $\phi_5$  a  $\phi_6$*

## 8. Řídící algoritmus

### 8.1. Grafické uživatelské rozhraní



Obr. 78: GUI – úvodní obrazovka – pravá ruka



Obr. 79: GUI – řízení levé ruky

Ke tvorbě grafického uživatelského rozhraní byla využita knihovna Tkinter. Výše zobrazené GUI se skládá z několika částí. Na levé straně jsou informační a řídicí panely. Pravá strana slouží k zobrazení aktuálního snímku z kamery a k vykreslování zvoleného grafu řízení – aktuální polohy z potenciometrů a polohy estimované z kamery. Z hlediska struktury programu je rozhraní rozděleno na šest tříd, které jsou podrobněji popsány v následujících podbodech.

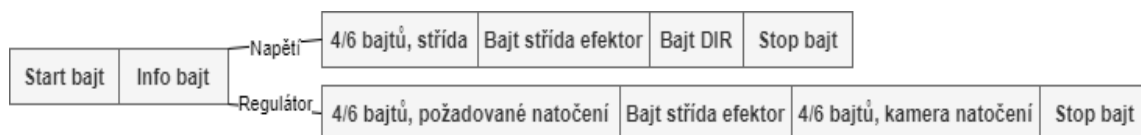
1. *Status frame* – vlevo nahoře – zobrazuje, zda je aktivní detekce míčků a nad kterou rukou se kamera nachází. Prostor pro rozšíření o další dvě kontrolky.
2. *Mode frame* – pod *Status frame* – slouží k volbě režimu ovládání. Lze přepnout mezi přímým nastavením napětí (*PWM*) a mezi regulací pomocí PI regulátorů (*Val*), implementovaných v Arduinu. V případě zvolení PI regulace se umožní volba mezi regulací pomocí potenciometrů a pomocí kamery.
3. *Control frame* – pod *Mode frame* – umožňuje ovládání všech motorů na ruce, nad kterou je kamera umístěna. Typ řízení se přepíná v závislosti na parametrech nastavených v *Mode frame*. Konkrétně je to buď napětí od -12 do 12 V, nebo úhlové natočení v definovaném rozsahu, vyplývajícím z fyzické konstrukce. Nastavení otevření efektoru se udává podle požadovaného vysunutí lineárního pohonu, a to od 0 do 30 mm. Lze zvolit i pohon, který se má vykreslovat v grafu. V případě zvolení vizuální zpětné vazby se zde volí, které konkrétní pohony jsou jí ovládány.
4. *On/off frame* – vlevo dole – slouží k zapínání a vypínání funkcionalit, a to detekce míčků a estimování úhlů natočení, zobrazování aktuálního snímku z kamery, vykreslování požadovaného grafu. Další tlačítko umožňuje opětovně detekovat QR kód a poslední tlačítko slouží k exportu získaných dat do .txt souborů – úhlových natočení z kamery, úhlových natočení z Arduina, požadovaných hodnot natočení a akčních zásahů.
5. *Image frame* – vpravo nahoře – při zapnutí v *On/off frame* zobrazuje RGB obraz z kamery, doplněný o pozitivní nálezy míčků. Obnovovací frekvence je omezena na 10 FPS, aby se snížila výpočetní náročnost programu.
6. *Graph frame* – vpravo dole – při zapnutí v *On/off frame* vykresluje grafy, zvolené v *Control frame*. Je rozdělen na dva subploty, na horním je červeně vykreslována křivka odhadu úhlu z kamery v radiánech, zeleně křivka získaná odečítáním napětí z potenciometrů. Spodní subplot zobrazuje akční zásah ve voltech. Pokud je zvoleno řízení nastavením napětí, tak přímo z aktuálního posuvu v *Control frame*. Pro PI regulátory je to akční zásah, který posílá Arduino do Pythonu ve zprávě.

## 8.2. Sériová komunikace

Počítač s Arduinem komunikuje přes USB pomocí UARTu. Rychlost komunikace je nastavena na 500 000 baudů za sekundu, což je nejvyšší hodnota, při které byla komunikace spolehlivá.

### 8.2.1. Formát zprávy

Zpráva posílaná z Pythonu má formát závislý na okolnostech. Začíná vždy start bajtem (<) a končí stop bajtem (>). První bajt v sobě nese informace, které dávají Arduinu pokyny k volbě ovládacího režimu (viz tab. 26). Následuje série dalších bajtů. Sekvence je analogická pro obě ruce, pouze se liší počtem bajtů, protože na pravé jsou dva pohony navíc.

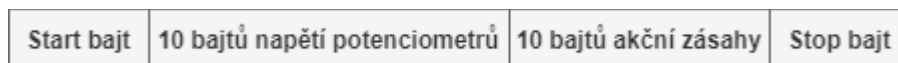


Obr. 80: Formát zprávy z Pythonu

Pořadí bitu	Význam bitu	Hodnoty bitu
0	Zvolená ruka	0 ... levá. 1 ... pravá
1	Ovládací režim	0 ... napětí. 1 ... regulátory
2	Vizuální zpětná vazba motor 1	0 ... vypnuta. 1 ... zapnuta
3	Vizuální zpětná vazba motor 2	0 ... vypnuta. 1 ... zapnuta
4	Vizuální zpětná vazba motor 3	0 ... vypnuta. 1 ... zapnuta
5	Vizuální zpětná vazba motor 4	0 ... vypnuta. 1 ... zapnuta
6	Vizuální zpětná vazba motor 5 <sup>1</sup>	0 ... vypnuta. 1 ... zapnuta
7	Vizuální zpětná vazba motor 6 <sup>1</sup>	0 ... vypnuta. 1 ... zapnuta

Tab. 26: Význam bitů prvního bajtu

Zpráva z Arduina má ve všech případech stejný formát. Nejdříve je poslán start bajt, následně deset bajtů s napětími na jednotlivých potenciometrech, reprezentujících natočení pohonů. Poté akční zásahy všech deseti pohonů a na závěr stop bajt.



Obr. 81: Formát zprávy z Arduina

### 8.2.2. Script v Pythonu

K realizaci komunikace slouží v Pythonu vlastní třída, která obsahuje všechny potřebné informace a čerpá nastavení ze zbylých tříd. Během inicializace se spustí vlákno podle toho, nad kterou rukou se kamera nachází. Vlákni poté cyklicky posílá a přijímá zprávy. Kromě toho také přepočítává přijatá data z potenciometrů na odpovídající kloubová natočení podle vzorce (44) a přijatá data o napětí na napětí podle vzorce (45).

$$\varphi_i = \frac{3\pi}{2} \cdot \frac{byte_i - offset_i}{255} \cdot sig_i \quad (44)$$

, kde  $\varphi_i$  ... natočení i-tého pohonu [rad]  
 $byte_i$  ... hodnota i-tého přijatého bytu [-]  
 $offset_i$  ... bytová hodnota, na které je natočení i-tého pohonu nulové [-]  
 $sig_i$  ... směr kladné výchylky natočení, -1, nebo 1

$$U_i = \frac{24}{12} \cdot byte_i + 12 \quad (45)$$

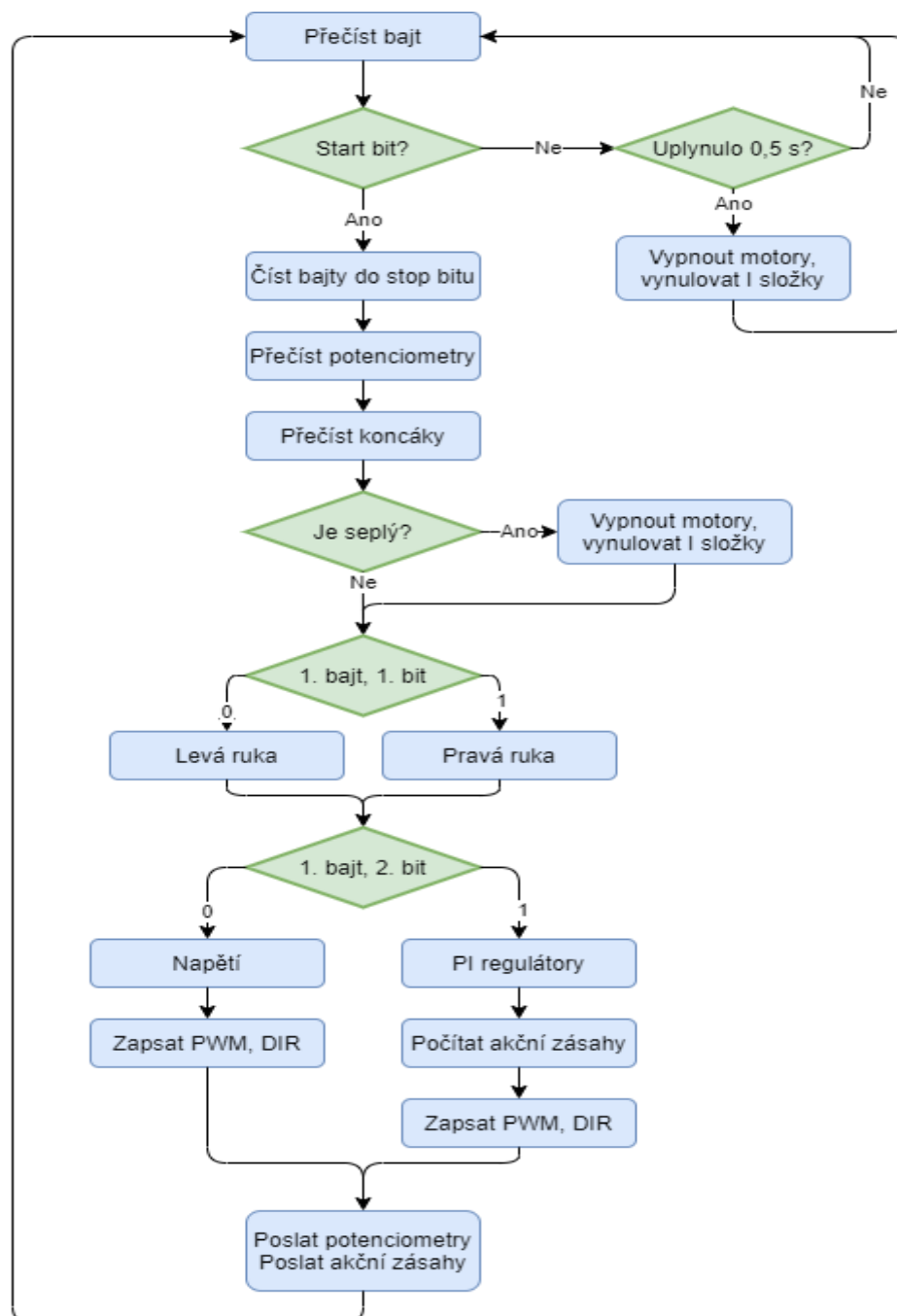
, kde  $U_i$  ... akční zásah i-tého pohonu [V]  
 $byte_i$  ... hodnota i-tého přijatého bytu [-]

<sup>1</sup> Platí pouze pro pravou ruku



### 8.2.3. Script v Arduinu

Script v Arduinu má za úkol vykonávat úkony podle dat, která přijal přes UART. Zjednodušený cyklus je popsán obrázkem níže. Kód se větví podle informací z prvního přijatého bajtu. Podle toho zapisuje napětí přímo na piny, nebo vypočítává akční zásahy regulátorů. V každém cyklu odečítá data z potenciometrů a kontroluje, zdali není sepnut koncový spínač. V tom případě okamžitě vypíná všechny pohony a vynuluje akční zásahy i I složky.



Obr. 82: Flowchart Arduino kódu

## 9. Řízení vizuální zpětnou vazbou

Vizuální zpětnou vazbu, vytvořenou v předešlých kapitolách, nyní aplikujeme na řízení jednotlivých pohonů. Spouští se zaškrtnutím odpovídajícího pole v grafickém uživatelském rozhraní a jde vybrat, které konkrétní pohony jí budou řízeny. Práce [1] využívá pro všechny pohony diskrétní PI regulátory dané rovnicí (46), takže se převzaly a upravily pro potřeby této práce. Experimentálně se řešil problém s délkou filtračního okna pro *moving average* filtr. Maximální délka  $x$ , při které nedochází k žádným oscilacím, způsobeným zpožděním, je  $x = 4$ .

$$u(nT) = k_p \cdot e(nT) + k_I \cdot T \sum_{k=0}^{n-1} e(kT) \quad (46)$$

, kde  $n$  je pořadí vzorku,  
 $T$  je perioda cyklu,  
 $u$  je akční zásah,  
 $k_p$  je proporční složka regulátoru,  
 $e$  je regulační odchylka,  
 $k_I$  je integrační složka regulátoru.

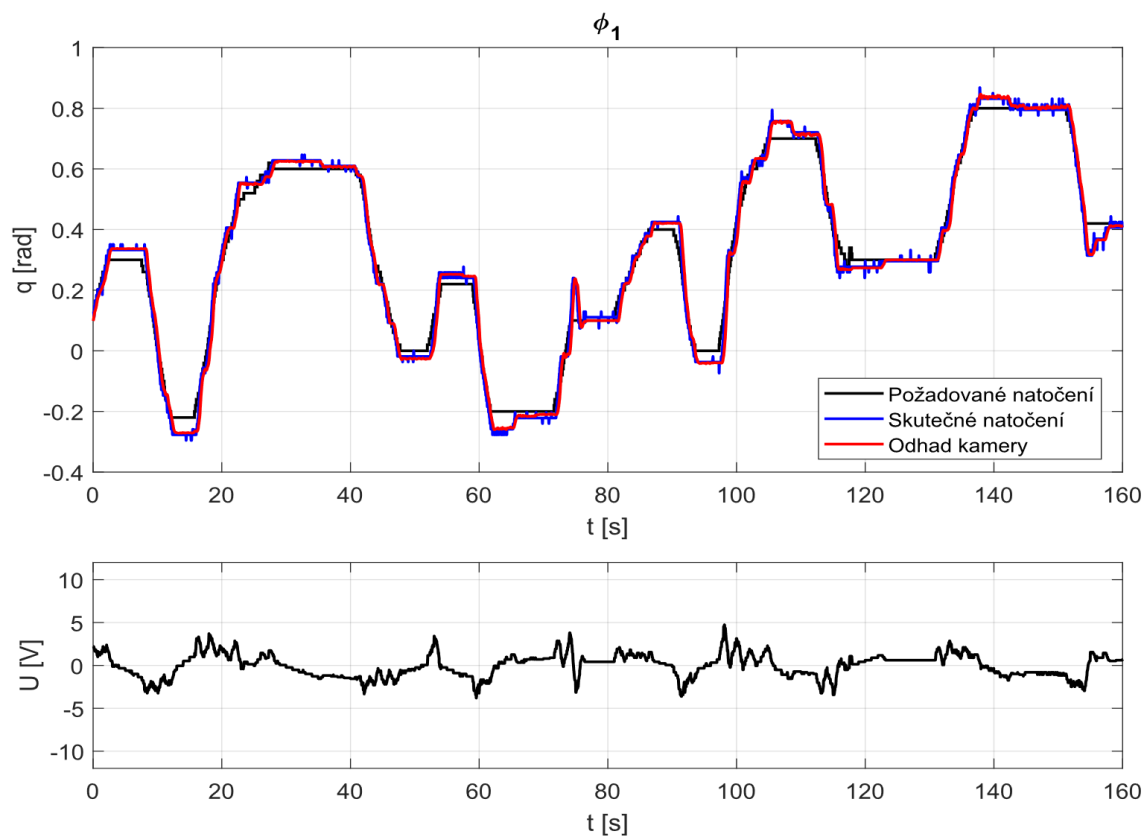
Jelikož lepší naladění parametrů všech regulátorů je předmětem práce [1], tak výsledné hodnoty posloužily jako inspirace pro volbu koeficientů regulátorů v této práci. V práci [1] běží program na 30 Hz, toho však v této práci docílit nejde, protože kamera má snímací frekvenci 30 FPS a pokud započteme další výpočetní čas a zohledníme-li výkon notebooku, na kterém byl kód spouštěn, pohybuje se výsledná frekvence okolo 20–25 Hz. Z tohoto důvodu je žádoucí regulátory mírně oslabit, protože s původními parametry by mohly být nestabilní.

Pohon	I1	I2	I3	I4	II1	II2	II3	II4	II5	II6
P složka	24	80	25	30	20	80	23	60	40	20
I složka	10	5	10	0	7	20	3	0	10	6

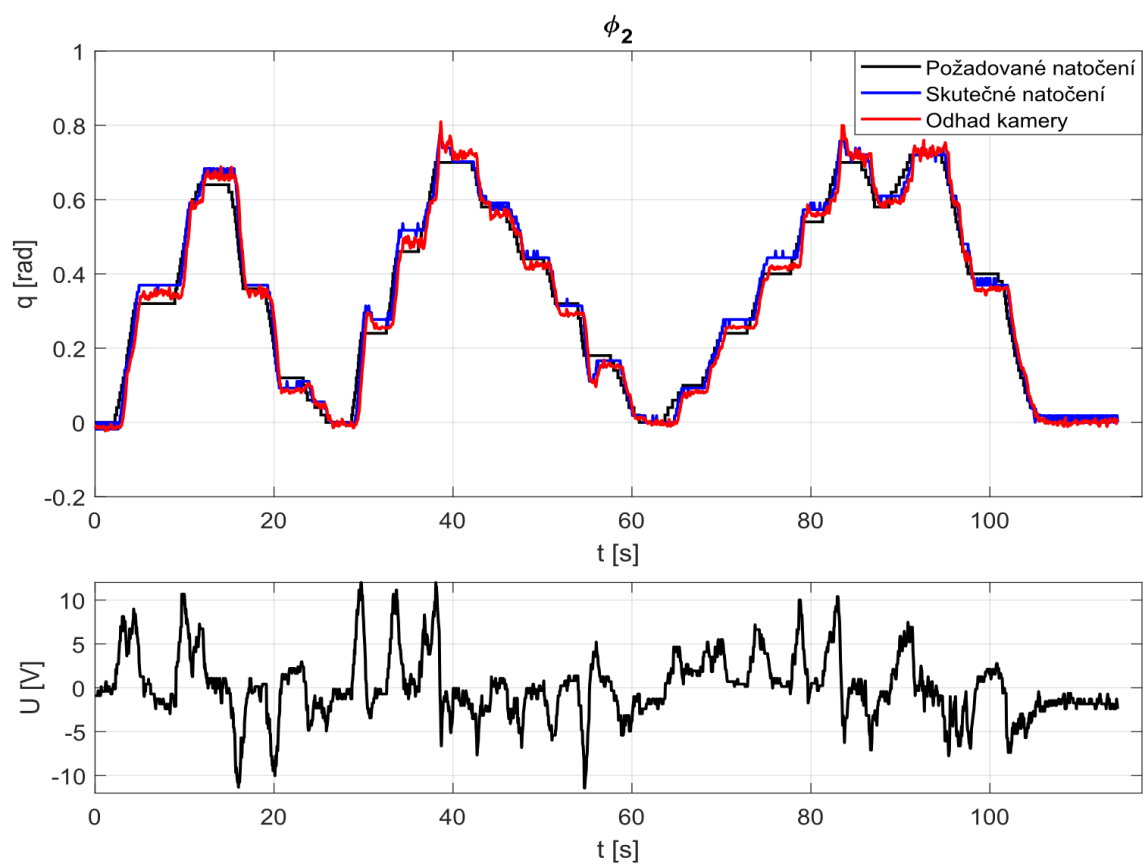
Tab. 27: Parametry regulátorů

### 9.1. Výsledek

Pomocí vizuální zpětné vazby lze řídit všechny pohony až na jeden. II6 motor se uřídit nepodařilo, protože kvůli krátkému rameni byla chyba určování natočení příliš velká. Nejlepších výsledků bylo dosaženo u motorů s nižšími pořadovými čísly, pro které lze vizuální zpětnou vazbu považovat za plnohodnotnou náhradu potenciometrů. Na grafech níže jsou zobrazeny dva ukázkové grafy, jeden z každé ruky. Zbytek grafů se nachází na konci této práce v přílohách (obr. 85–93).



Obr. 83: Vizuální zpětnovazební řízení II



Obr. 84: Vizuální zpětnovazební řízení II2

## 10. Závěr

Předmětem této práce byla kompletní realizace laboratorního modelu robotického manipulátoru. Autor v úzké spolupráci s kolegou [1] postavil „humanoidního“ robota s dvěma pažemi. Jedním z prvotních požadavků byla nízká cena, která se ve finále pohybovala okolo dvaceti tisíc korun, přičemž největší položku představovala elektronika, zejména Arduino Mega 2560, H můstky Cytron, proudové senzory a svorky na DIN lištu. Naopak mechanickou část se podařilo pořídit velmi levně. Motory poskytl zdarma strýc autora, hliníkové profily a díly jsou ze second-hand obchodu, pásovina z oceli je též drobnou položkou. Ostatní díly byly nalezeny v Mechlabu nebo u autorů doma a plastové díly byly vytištěny na 3D tiskárně.

Výsledkem je robot, který má dvě funkční ruce, z nichž jedna má čtyři stupně volnosti a druhá šest. Tato práce se podrobněji zabývala rukou se čtyřmi stupni volnosti, která je inspirována paletovacími roboty. Je poháněna čtyřmi DC motory z automobilů a navrhnutý efektor se otevírá a zavírá díky lineárnímu pohonu. Celé tělo robota je schopno vertikálního pohybu díky lineárnímu vedení a heverovému zvedáku, poháněnému motorem z předních střeračů. Robot je umístěn na základně na kolečkách, což značně usnadňuje manipulaci s ním. Konstrukce obsahuje nedostatky, jako například vůle v pohonech a konstrukční nepřesnosti, v důsledku omezených možností výroby dílů. Tento aspekt však vzhledem k velkému důrazu na cenu není negativem, protože robot svůj účel plní.

Dalším cílem bylo nahradit na obou rukou zpětnou vazbu z potenciometrů natočeními určenými pomocí kamery. Úkol se řešil umístěním míček na vhodná místa na konstrukci, jejich poloha v prostoru sloužila k počítání natočení algebraickými rovnicemi. Podařilo se takto řídit všechny motory až na jeden, protože jeho krátké rameno v kombinaci se šumem zvyšovalo chybu na nepříjemnou velikost. Kvalitu metody omezují zejména vlastnosti kamery (šum, nepřesnost měření) a vlastnosti konstrukce (nepřesnosti v uloženích). Přesto zejména pro motory blíže k základně byla vizuální zpětná vazba schopna potenciometry nahradit s minimálním snížením kvality regulace.

Místo zadané ukázkové úlohy se předkládá grafické uživatelské rozhraní, kterým lze ovládat robota různými způsoby. Jmenovitě napětím, regulátorem s potenciometrem a regulátorem s vizuální zpětnou vazbou. Bylo vytvořeno v době nouzového stavu, vyhlášeného kvůli epidemii COVID-19, protože se robot nacházel u kolegy na Slovensku a nevědělo se, kdy se autor k robotovi dostane a jestli bude mít čas na realizaci ukázkové úlohy.

Do budoucna by tento výtvar mohl sloužit pro realizaci dalších bakalářských či diplomových prací. Nabízí se různá vylepšení, z nichž některé nápady jsou v bodech níže:

1. Finalizace designu konstrukce včetně zakrytování
2. Návrh a instalace hlavy
3. Výměna pohonů za kvalitnější a zlepšení konstrukce
4. Realizace jiných metod vizuální zpětné vazby – např. umístění kamery k efektoru
5. Implementace dalších funkcí – detekce překážek, detekce objektu a jeho uchopení
6. Rozšíření GUI o další funkce – inverzní kinematika, řízení heveru

## **11. Seznam použitých zkratek**

**DC** Direct Current

**DoF** Degree of Freedom

**GND** Ground

**GUI** Graphic User Interface

**PLC** Programmable Logic Controller

**PWM** Pulse Width Modulation

**ToF** Time of Flight

**UART** Universal Asynchronous Receiver-Transmitter

**USB** Universal Serial Bus

## 12. Zdroje

- [1] CHLAŇ, Jakub. *Senzorika a řízení pohybu pro humanoidního robota* [online]. Brno, 2020 [cit. 2020-06-19]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/125325>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Jan Najman.
- [2] Řada robotů FANUC M-410. In: *Fanuc* [online]. [cit. 2020-06-02]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/stránka-filtru-robotů/řada-m-410>
- [3] M-410iC/110. In: *FANUC* [online]. [cit. 2020-06-02]. Dostupné z: <https://www.fanuc.eu/~media/files/pdf/products/robots/flyer/mfl-03071-ro%20m-410ic110%20introduction/m-410ic110-flyer-en.pdf?la=cs>
- [4] Paletovací roboty firmy KUKA. In: *KUKA* [online]. [cit. 2020-06-02]. Dostupné z: <https://www.kuka.com/cs-cz/produkty,-služby/robotické-systémy/průmyslové-roboty/paletovací-robot>
- [5] KR 120 R3200 PA. In: *KUKA* [online]. [cit. 2020-06-02]. Dostupné z: [https://www.kuka.com/~media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000197324\\_en.pdf](https://www.kuka.com/~media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000197324_en.pdf)
- [6] Tinkerkit Braccio robot. In: *Arduino Official Store* [online]. [cit. 2020-06-02]. Dostupné z: <https://store.arduino.cc/tinkerkit-braccio-robot>
- [7] BTS7960 High Current 43A H-Bridge Motor Driver. In: *Handson Technology* [online]. [cit. 2020-06-04]. Dostupné z: <https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%20Driver.pdf>
- [8] DC Motor Driver. In: *Cytron* [online]. [cit. 2020-06-04]. Dostupné z: <https://www.cytron.io/c-dc-motor-driver>
- [9] Modul 2x H-můstek s L298N. In: *GM Electronic* [online]. [cit. 2020-06-04]. Dostupné z: <https://www.gme.cz/modul-2x-h-mustek-s-l298n>
- [10] PLC - Programovatelný logický automat. In: *PLC automatizace* [online]. [cit. 2020-06-04]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc.htm>
- [11] Logický modul LOGO!. In: *Siemens* [online]. [cit. 2020-06-04]. Dostupné z: <https://new.siemens.com/cz/cs/products/automation/systems/industrial/plc/logo.html>
- [12] How to Design a PCB Layout. In: *Circuit Basics* [online]. [cit. 2020-06-18]. Dostupné z: <https://www.circuitbasics.com/make-custom-pcb/>
- [13] Arduino - Introduction. In: *Arduino* [online]. [cit. 2020-06-04]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>

- [14] Arduino Nano. In: *Arduino Official Store* [online]. [cit. 2020-06-04]. Dostupné z: <https://store.arduino.cc/arduino-nano>
- [15] Arduino Mega 2560 Rev3. In: *Arduino Official Store* [online]. [cit. 2020-06-04]. Dostupné z: <https://store.arduino.cc/arduino-mega-2560-rev3>
- [16] Tales In Tech History: Microsoft Kinect. In: *Silicon UK* [online]. [cit. 2020-06-05]. Dostupné z: <https://www.silicon.co.uk/e-innovation/microsoft-kinect-history-226781>
- [17] Azure Kinect DK. In: *Microsoft* [online]. [cit. 2020-06-05]. Dostupné z: <https://azure.microsoft.com/cs-cz/services/kinect-dk/>
- [18] Intel® RealSense™ Depth Camera SR305. In: *Intel RealSense* [online]. [cit. 2020-06-05]. Dostupné z: <https://www.intelrealsense.com/depth-camera-sr305/>
- [19] Which Intel RealSense device is right for you?. In: *Intel RealSense* [online]. [cit. 2020-06-05]. Dostupné z: <https://www.intelrealsense.com/which-device-is-right-for-you/>
- [20] Intel® RealSense™ Depth Camera D435. In: *Intel RealSense* [online]. [cit. 2020-06-05]. Dostupné z: <https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435.html>
- [21] Astra Series. In: *Orbbec* [online]. [cit. 2020-06-17]. Dostupné z: <https://orbbec3d.com/product-astra-pro/>
- [22] RGB and CMYK Colour systems. In: *Poetic Mind* [online]. [cit. 2020-06-10]. Dostupné z: <https://www.poeticmind.co.uk/research/rgb-cmyk-colour-systems/>
- [23] HSC Color Space. In: *Ronja's Shader Tutorials* [online]. 2019 [cit. 2020-06-10]. Dostupné z: <https://www.ronja-tutorials.com/2019/04/16/hsv-colorspace.html>
- [24] Hough Lines Transform Explained. *Medium* [online]. [cit. 2020-06-18]. Dostupné z: <https://medium.com/@tomasz.kacmajor/hough-lines-transform-explained-645feda072ab>
- [25] CHAUMETTE, François, S HUTCHINSON a Eric MARCHAND. Visual servo control, Part II: Advanced approaches. *IEEE Robotics and Automation Magazine* [online]. Institute of Electrical and Electronics Engineers, 2007, **14**(1), 109-118 [cit. 2020-06-18]. ISSN 1070-9932. Dostupné z: <https://hal.inria.fr/inria-00350638>
- [26] HUTCHINSON, S, G.D HAGER a P.I CORKE. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* [online]. IEEE, 1996, **12**(5), 651-670 [cit. 2020-06-18]. DOI: 10.1109/70.538972. ISSN 1042-296X. Dostupné z: <https://ieeexplore.ieee.org/document/538972>
- [27] KIRK, A.G, J.F O'BRIEN a D.A FORSYTH. Skeletal parameter estimation from optical motion capture data. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* [online]. San Diego: IEEE,



2005, **2**, 782-788 vol. 2 [cit. 2020-06-18]. DOI: 10.1109/CVPR.2005.326. ISBN 0769523722. ISSN 10636919.

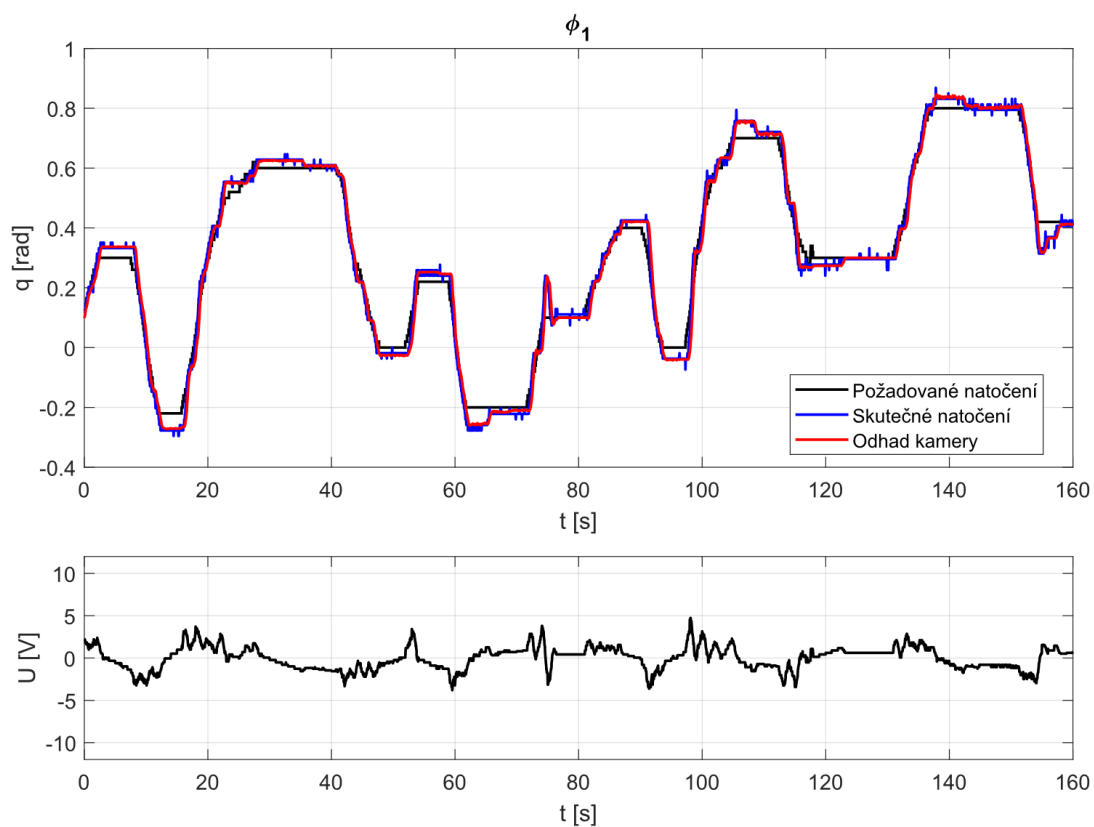
- [28] Actuonix L12 Linear Actuator 30mm 210:1 12V Analog. In: *RobotShop* [online]. [cit. 2020-06-14]. Dostupné z: <https://www.robotshop.com/en/actuonix-l12-linear-actuator-30mm-2101-12v-analog.html>
- [29] MakeBlock Robot Gripper. In: *RobotShop* [online]. [cit. 2020-06-18]. Dostupné z: <https://www.robotshop.com/en/makeblock-robot-gripper.html>
- [30] Analogový multiplexer CD74HC4067. In: *Arduino návody* [online]. [cit. 2020-06-15]. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/analogovy-multiplexer-cd74hc4067.html>
- [31] TCA9548A I2C Multiplexer. In: *Adafruit* [online]. [cit. 2020-06-15]. Dostupné z: <https://www.adafruit.com/product/2717>
- [32] Arduino Mega Tutorial [Pinout]. In: *DiyIoT* [online]. [cit. 2020-06-15]. Dostupné z: <https://diyi0t.com/arduino-mega-tutorial/>

## 13. Přílohy

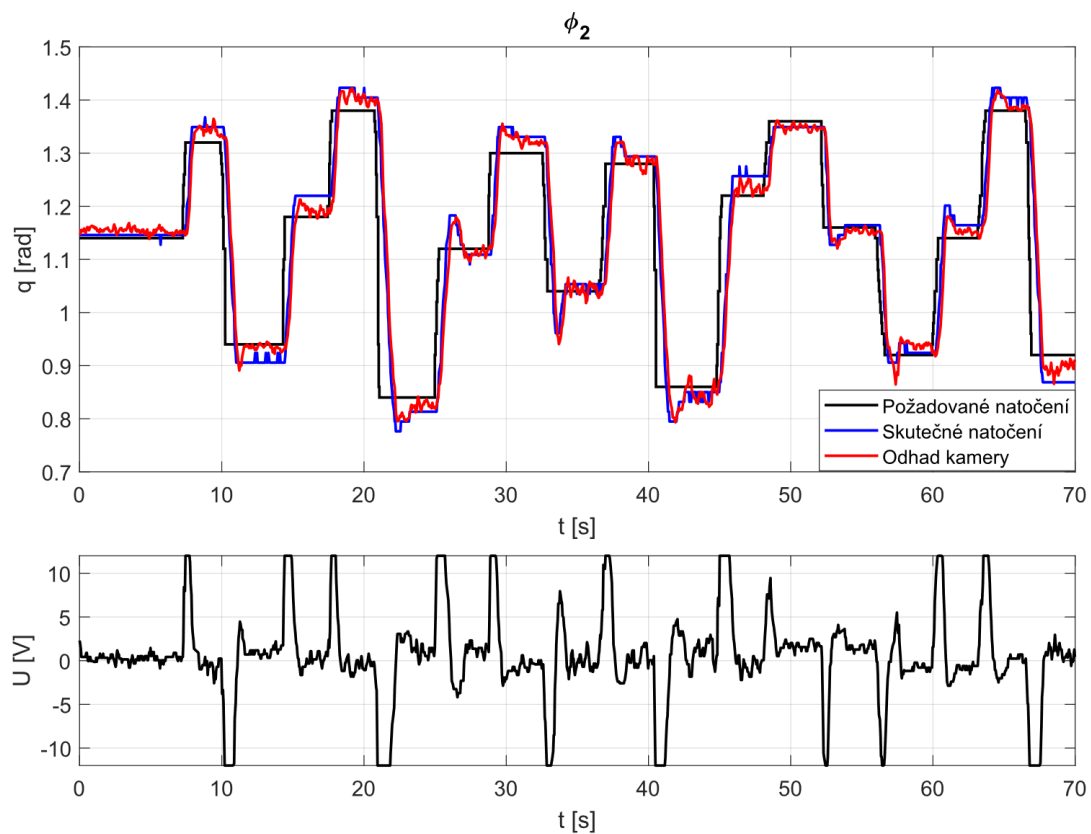
### 13.1. Elektronické

- Zdrojový kód pro Arduino
- Zdrojové kódy pro Python: GUI; test RGB, HSV, Houghova transformace
- Modely dílů v prostředí Autodesk Fusion 360

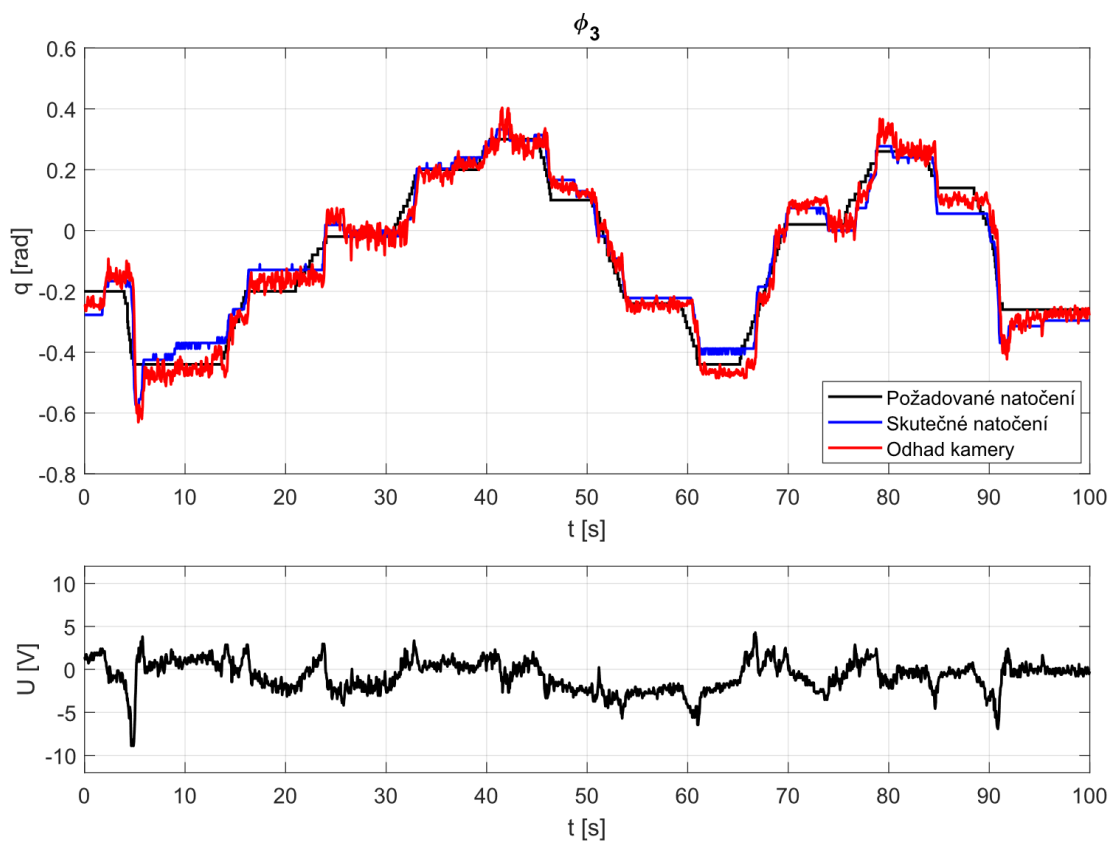
### 13.2. Grafické



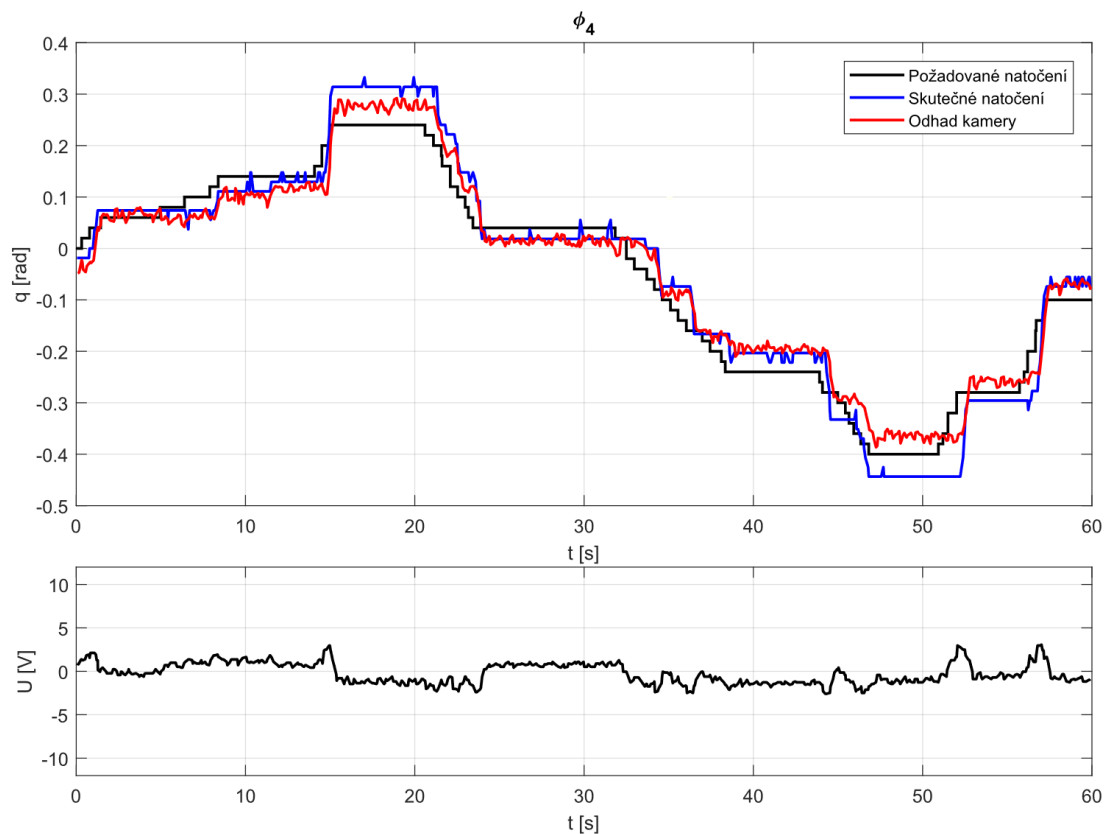
Obr. 85: Vizuální zpětnovazební řízení II



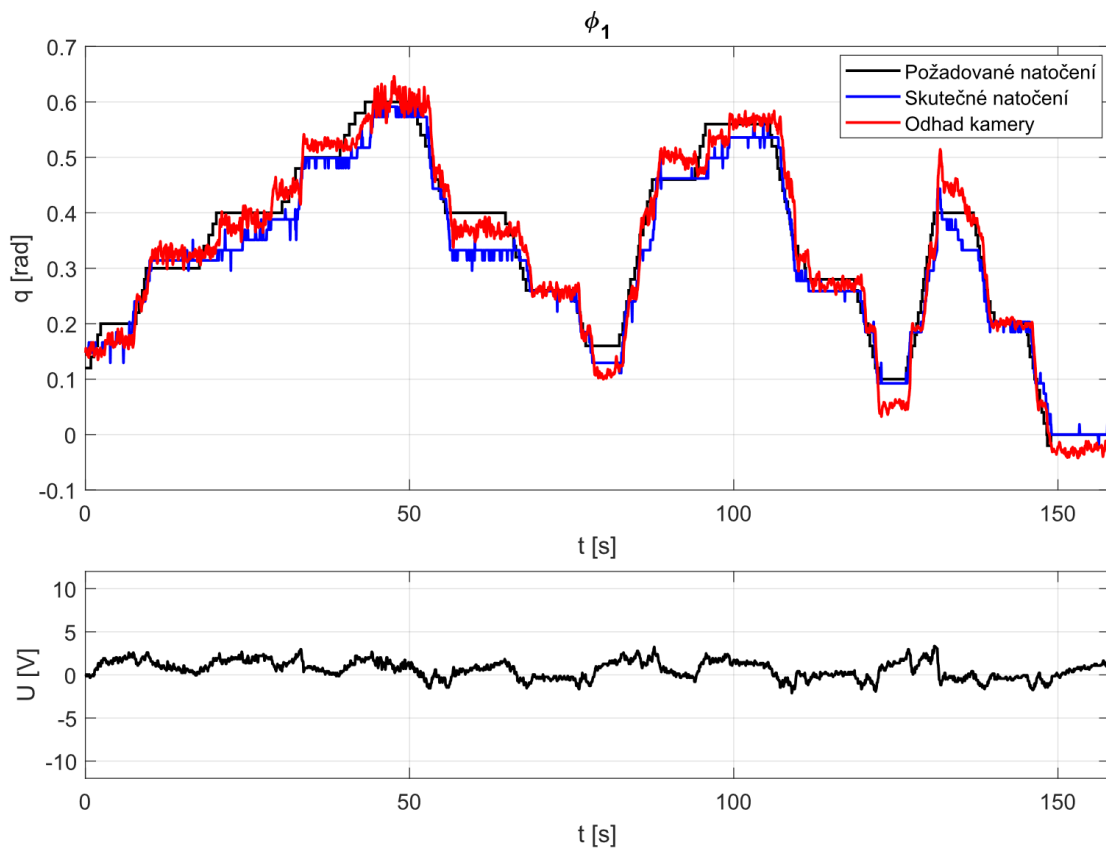
Obr. 86: Vizualní zpětnovazební řízení I2



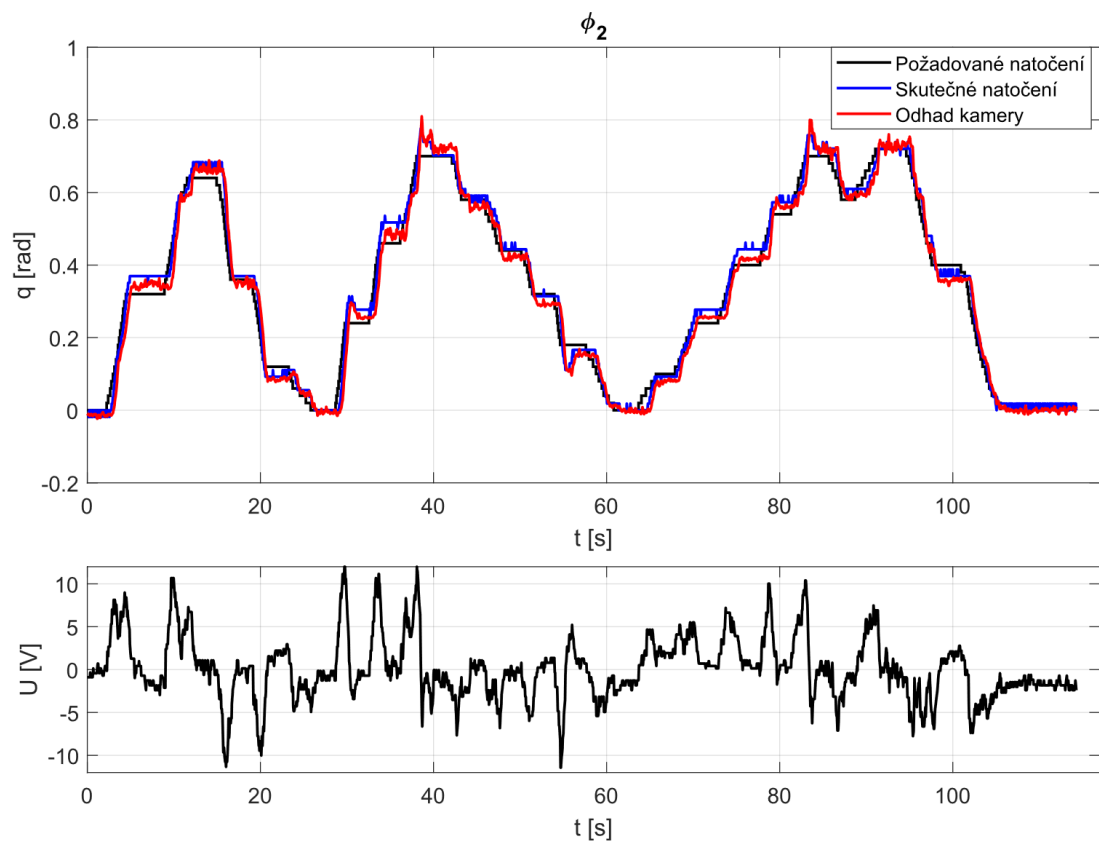
Obr. 87: Vizualní zpětnovazební řízení I3



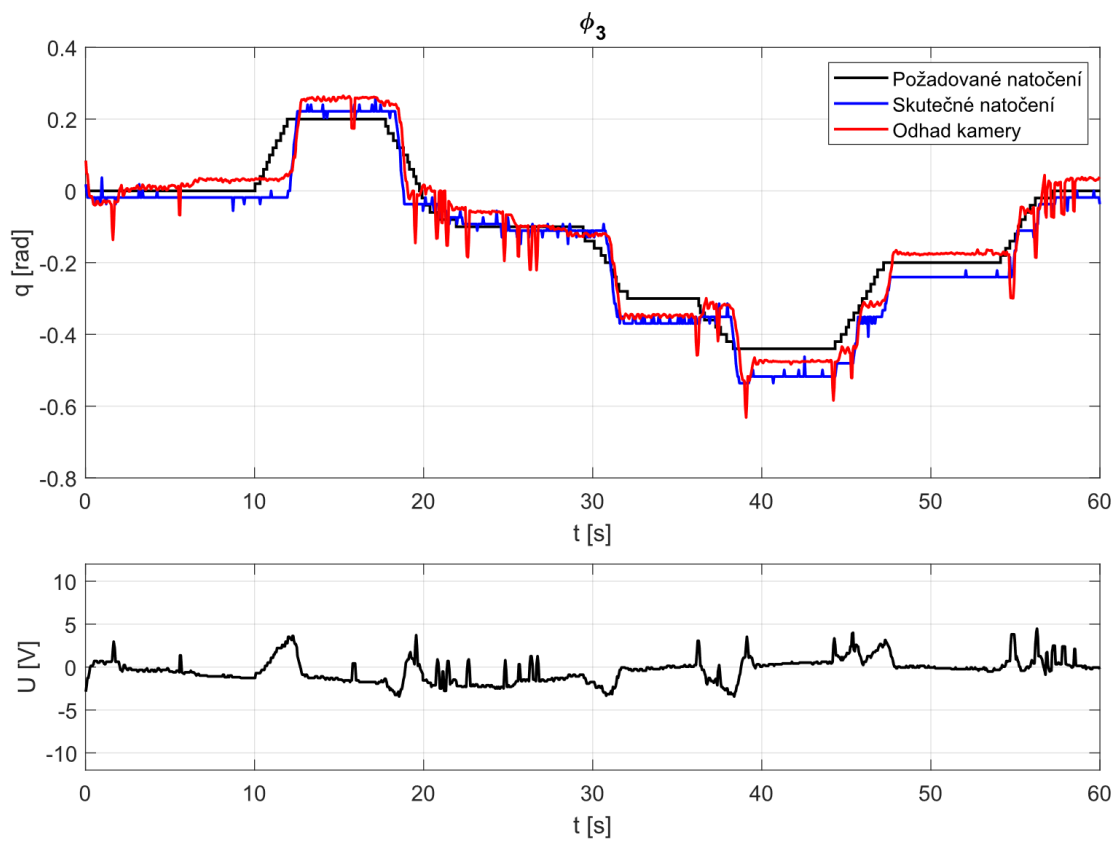
Obr. 88: Vizuální zpětnovazební řízení I4



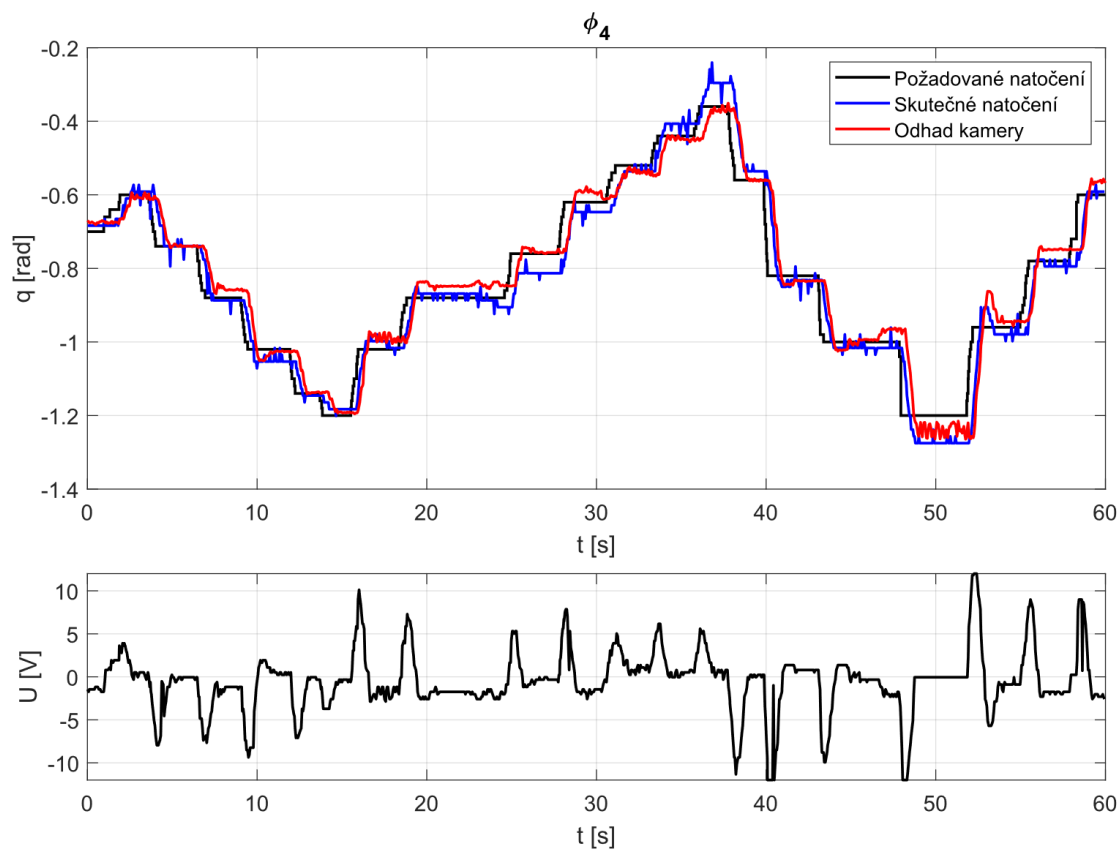
Obr. 89: Vizuální zpětnovazební řízení III



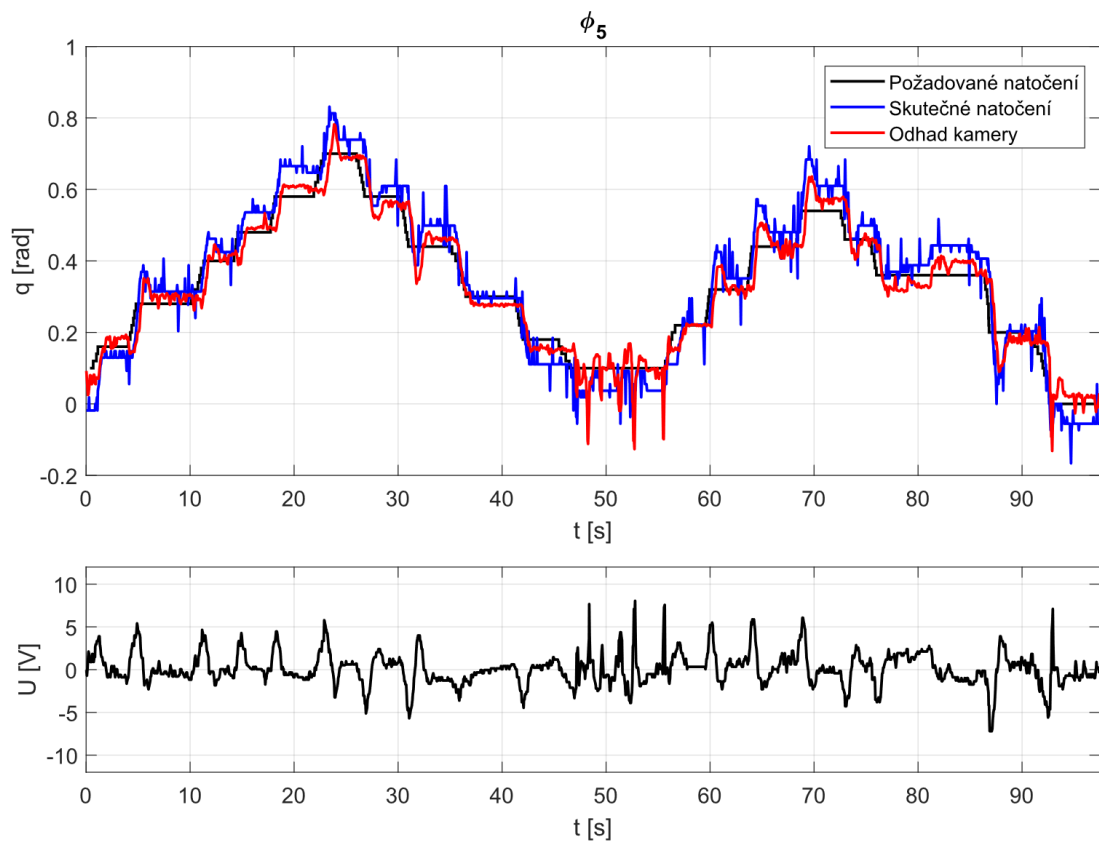
Obr. 90: Vizuální zpětnovazební řízení II2



Obr. 91: Vizuální zpětnovazební řízení II3



Obr. 92: Vizuální zpětnovazební řízení II4



Obr. 93: Vizuální zpětnovazební řízení II5